Consultative
Committee for
Space Data Systems

DRAFT RECOMMENDATION FOR SPACE DATA SYSTEM STANDARDS

SPACE LINK EXTENSION—SERVICE MANAGEMENT SPECIFICATION

CCSDS 910.5-R-2

RED BOOK

September 2001

AUTHORITY

Issue:	Red Book, Issue 2
Date:	September 2001
Location:	N/A

(WHEN THIS RECOMMENDATION IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF AUTHORITY)

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS Recommendations is detailed in reference [E1], and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the address below.

This Recommendation is published and maintained by:

CCSDS Secretariat
Program Integration Division (Code MT)
National Aeronautics and Space Administration
Washington, DC 20546, USA

STATEMENT OF INTENT

(WHEN THIS RECOMMENDATION IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF INTENT)

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of member space Agencies. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **Recommendations** and are not considered binding on any Agency.

This **Recommendation** is issued by, and represents the consensus of, the CCSDS Plenary body. Agency endorsement of this **Recommendation** is entirely voluntary. Endorsement, however, indicates the following understandings:

– Whenever an Agency establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommendation**. Establishing such a **standard** does not preclude other provisions which an Agency may develop.

– Whenever an Agency establishes a CCSDS-related standard, the Agency will provide other CCSDS member Agencies with the following information:

  - The **standard** itself.
  - The anticipated date of initial operational capability.
  - The anticipated duration of operational service.

– Specific service arrangements are made via memoranda of agreement. Neither this Recommendation nor any ensuing standard is a substitute for a memorandum of agreement.

No later than five years from its date of issuance, this **Recommendation** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or, (3) be retired or canceled.

In those instances when a new version of a **Recommendation** is issued, existing CCSDS-related Agency standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each Agency to determine when such standards or implementations are to be modified. Each Agency is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommendation.

FOREWORD

(WHEN THIS RECOMMENDATION IS FINALIZED, IT WILL CONTAIN THE FOLLOWING FOREWORD)

This document is a technical **Recommendation** for use in developing ground systems for space missions and has been prepared by the **Consultative Committee for Space Data Systems** (CCSDS). The Space Link Extension Service Management described herein is intended for missions that are cross supported between Agencies of the CCSDS.

This **Recommendation** establishes the common aspects of Service Management for the specification of data services that extend the space to ground communication services previously defined by CCSDS. It allows implementing organizations within each Agency to proceed coherently with the development of compatible derived Standards for the ground systems that are within their cognizance. Derived Agency Standards may implement only a subset of the optional features allowed by the **Recommendation** and may incorporate features not addressed by the **Recommendation**.

Through the process of normal evolution, it is expected that expansion, deletion or modification to this document may occur. This Recommendation is therefore subject to CCSDS document management and change control procedures as defined in reference [E1]. Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this document should be addressed to the CCSDS Secretariat at the address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- British National Space Centre (BNSC)/United Kingdom.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- National Aeronautics and Space Administration (NASA)/USA.
- National Space Development Agency of Japan (NASDA)/Japan.
- Russian Space Agency (RSA)/Russian Federation.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- Centro Tecnico Aeroespacial (CTA)/Brazil.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Communications Research Centre (CRC)/Canada.
- Communications Research Laboratory (CRL)/Japan.
- Danish Space Research Institute (DSRI)/Denmark.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Federal Service of Scientific, Technical & Cultural Affairs (FSST&CA)/Belgium.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space and Astronautical Science (ISAS)/Japan.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- MIKOMTEK: CSIR (CSIR)/Republic of South Africa.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- National Oceanic & Atmospheric Administration (NOAA)/USA.
- National Space Program Office (NSPO)/Taipei.
- Swedish Space Corporation (SSC)/Sweden.
- United States Geological Survey (USGS)/USA.

PREFACE

This document is a draft CCSDS Recommendation. Its Red Book status indicates that the CCSDS believes the document to be technically mature and has released it for formal review by appropriate technical organizations. As such, its technical contents are not stable, and several iterations of it may occur in response to comments received during the review process.

Implementers are cautioned **not** to fabricate any final equipment in accordance with this document's technical content.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 910.5-R-1	Space Link Extension—Service Management Specification	December 1999	Original issue, superseded
CCSDS 910.5-R-2	Space Link Extension—Service Management Specification	September 2001	Current issue

CONTENTS

Figure

Table

1 INTRODUCTION

1.1 PURPOSE OF THIS RECOMMENDATION

1.1.1 BASELINE AND REFERENCE FOR SPACE LINK EXTENSION (SLE) SERVICE MANAGEMENT

This draft Recommendation establishes a framework for managing Space Link Extension (SLE) services. It provides a common basis for scheduling, monitoring, and controlling the SLE transfer services that an SLE system provides to a space mission.

This draft Recommendation specifies the management information required to provide SLE transfer services, the interfaces involved, and the necessary sequencing.

1.1.2 RELATION TO CROSS SUPPORT REFERENCE MODEL

The *Cross Support Reference Model* (reference [1] provides the framework for defining SLE Service specifications to be used to support space missions. It defines functional and management components of the common characteristics of SLE services, as well as the template for SLE Service specifications and the identification of the SLE transfer services.

The *Cross Support Reference Model* (reference [1]) establishes a model for space mission data exchange as illustrated in figure 1-1. The space mission users and management are represented by the Mission Data Operations System (MDOS), which exchanges return and forward data with the Space Element. The SLE system transfers this application data using SLE transfer services. In addition, the MDOS and the SLE system exchange management data for managing the SLE transfer services.

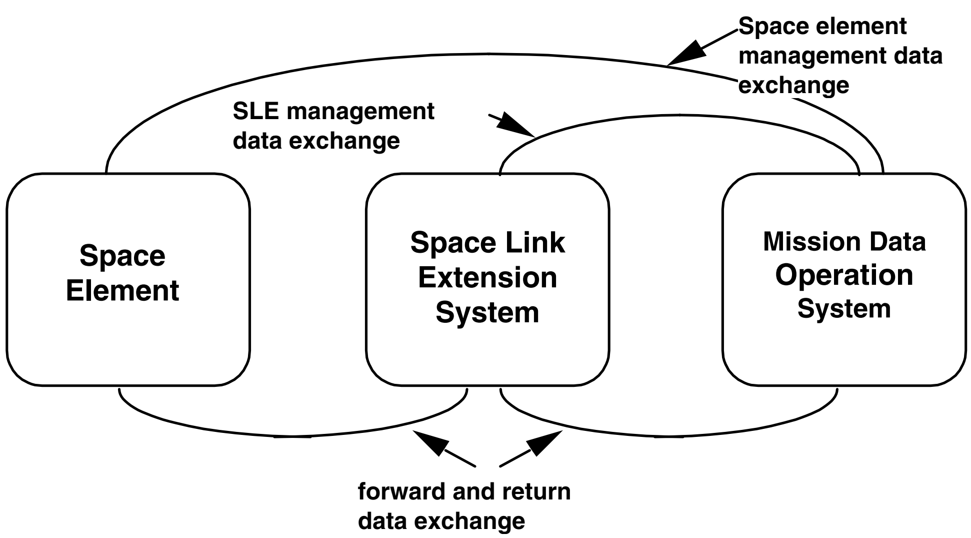


Figure 1-1: Space Mission Data Exchange

This draft *SLE Service Management Specification* complements SLE transfer service specifications developed within the framework of the *Cross Support Reference Model* (reference [1]). It specifies the means by which ground data systems can schedule, configure, operate, and monitor the production and provision of SLE transfer services for a space mission.

1.2 SCOPE

1.2.1 SLE SERVICE MANAGEMENT AND SLE SERVICES

SLE services comprise:

a) SLE Transfer Services, which are concerned with the ground part of the data transfer. This transfer is either within the ground data system or between the ground data system and ground data sinks.

b) SLE Service Management, which controls the scheduling and provision of the SLE Transfer Services by ground systems.

This document applies to the Service Management portion of SLE services.

1.2.2 SLE SERVICE MANAGEMENT AND OPEN SYSTEM INTERCONNECTION (OSI)

The Open System Interconnection (OSI) standard identifies five Management Functional Areas:

a) Configuration Management. Definition of SLE Transfer Service instances grouped into service packages, which identify input and output data channels, processing parameters and storage requirements.

b) Accounting Management. Reports of data received, sent, and processed.

c) Fault Management. Notification of faulty conditions that affect the services being provided.

d) Security Management. Administration of security credentials, control of access to interfaces, authentication of requests for data.

e) Performance Management. Not addressed in this document, as this aspect is not visible to the user. (*OSI Basic Reference Model,* reference [E5].)

This document uses managed objects to support these five areas. This document emphasizes the configuration management functional area and lays the foundation for security management. Additional management capabilities that support these functional areas are items for future study. These potential future capabilities include a method for the service user to report service problems to the provider (e.g., ‘trouble ticketing’).

1.2.3 ASSUMPTIONS

This draft Recommendation applies to the common Service Management aspects of SLE services. The following assumptions are made:

a) the context is that of a single space mission;

b) within this space mission a single spacecraft is considered;

c) this spacecraft's telemetry and telecommand are compliant with CCSDS Advanced Orbiting System (AOS), Packet Telemetry (PT), and Telecommand (TC) Recommendations (references [2] - [8]);

d) all ground end-users (i.e., data sinks or sources) are affiliated with a single mission utilization management entity;

e) mapping rules exist by which the operations defined in section 4 can be mapped onto equivalent facilities of CCSDS-recognized technologies.

NOTE – See *Mapping Rules* (reference [9]) for prototype mapping rules.

1.2.4 LIMITATIONS

The following points are not covered by this draft Recommendation:

a) Although sharing ground systems between multiple space missions or between multiple spacecraft of the same space mission is not explicitly modeled, this draft Recommendation in no way precludes sharing ground systems.

b) This document presupposes underlying communication capabilities, which may include security aspects. However, management of these capabilities is outside the scope of this draft Recommendation.

c) Ground systems and/or services that are not directly concerned with the transport of data compliant to CCSDS AOS, PT, and TC Recommendations are not described. Processing of data held within the data fields of Source Packet Protocol Data Units (PDU) is described in references [2]–[8], and is outside the scope of this draft Recommendation.

d) Mission planning, specifically the selection of an individual ground station to support a given Space Link session, is not within the scope of this draft Recommendation.

1.3 APPLICABILITY

1.3.1 APPLICABILITY OF THIS DRAFT RECOMMENDATION

This draft Recommendation serves as a guideline for the development of compatible internal Agency standards for SLE systems. Systems embraced by this draft Recommendation include manned and unmanned free-flying spacecraft and space transportation systems. This draft Recommendation is particularly relevant to SLE systems that are involved in cross support.

This and related SLE Recommendations apply to data systems that are able to:

a) receive CCSDS-conformant Space Link data structures from a spacecraft via a Space Link; or

b) send CCSDS-conformant Space Link data structures to a spacecraft via a Space Link; or

c) transfer such CCSDS-conformant Space Link data structures between ground-based entities.

1.3.2 LIMIT OF APPLICABILITY

This draft Recommendation is not intended to be a specification for actual implementations of SLE systems.

To achieve compatibility, the structure and behavior of independent implementations of SLE managed objects must adhere to the formal specification contained in the *SLE Service Management Managed Object Formal Specification* (reference [10]). It complements sections 5 through 14 of this draft Recommendation and presents a precise formal specification of the structure of the SLE managed objects.

The exchange of SLE management information could be implemented using a variety of technologies such as management services and protocols, middleware technologies, simple file transfers, and human procedures.

NOTES

1 Guidelines for Definition of Managed Objects (GDMO) and Unified Modeling Language (UML) notations are neutral with respect to implementation technologies. For example, the use of GDMO, which is developed for OSI management systems, does not imply that SLE implementations must use OSI management services and protocols.

2 At the time this draft Recommendation was developed, several CCSDS Agencies were investigating implementation solutions for SLE Complex Management such as GDMO/ASN.1, Common Object Request Broker Architecture (CORBA), Distributed Computing Environment (DCE), Java Remote Method Invocation (RMI), Distributed

Component Object Model (DCOM), and Internet Simple Network Management Protocol (SNMP).

3 Specific examples of implementation technologies include management services and protocols like Common Management Information Service (CMIS)/Common Management Information Protocol (CMIP) or Internet SNMP; technologies like CORBA, DCE, DCOM, Java RMI, and their related services; file transfers together with detailed specifications of structure and contents of each of the files; and procedures that are supported by form sheets and voice loop protocols.

Regardless of the implementation technologies used, it is essential that the interface between SLE Complex Management and SLE Utilization Management (these entities are defined in the *Cross Support Reference Model,* reference [1]) is implemented according to the *Mapping Rules* (reference [9]), in which a collection of mapping rules is specified for how SLE service management operations (see section 4) relate to features of each of the technologies recognized by the CCSDS. Reference [9] also specifies rules for constructing a gateway between recognized technologies.

If the SLE Complex Management and SLE Utilization Management are implemented using the same technologies, it is sufficient that the operation mapping rules be strictly applied.

If the SLE Complex Management and SLE Utilization Management are implemented using different technologies, it is necessary that the mapping rules be strictly applied by both entities, and that the appropriate construction rules be applied to the implementation of the gateway.

1.4 RATIONALE

The primary goal of CCSDS is to increase the level of interoperability among Agencies. This draft Recommendation furthers that goal by establishing the means to manage provision of SLE services to be used in the area where most cross support activity occurs: between the tracking stations or ground data handling systems of various Agencies and the mission specific components of a mission ground system. *Cross Support Concept* (reference [E3]) provides further discussion of the rationale for this draft Recommendation.

1.5 SLE SERVICE MANAGEMENT OBJECTIVES

The SLE Service Management concept is derived from the following goals:

a) The need to support space missions should be the driver of the management concept. Management concepts should be adopted to support the real needs of the mission activity.

b) The user of SLE transfer services should be isolated from the details of the individual Agency's SLE systems. The user will specify service characteristics, not the topology of the system.

c) A common management representation of services should be provided to users. The common representation will apply regardless of the number and combination of services offered by any particular Agency. It supports flexibility of an Agency to organize internal management domains as it sees fit, while presenting a consistent service view to the user, and it facilitates the Agencies' ability to hide system-specific details from users.

d) Adequate information should be provided to the user in order to provide sufficient accountability for all services performed by the system. The management concept does not specify internal management requirements or concepts for management data collection and reporting.

e) The aspects of the management process that lend themselves to automation and affect the costs of implementation are of primary concern.

1.6 HOW TO READ THIS DOCUMENT

This draft Recommendation is part of a document suite that addresses Space Data System cross support, particularly the SLE services. It is important for readers of this document to have a basic understanding of:

a) CCSDS Space Link and CCSDS Space Link Extension services;

b) Abstract Service Definition Conventions (ASDC);

c) Object-oriented Analysis description techniques;

d) industry-accepted Network Management techniques.

Readers of this document should proceed in the following manner:

a) Become generally familiar with the Abstract Service Definition Conventions described in *Standard Terminology, Conventions, and Methodology* document (reference [11]).

b) Become familiar with the CCSDS Space Link Extension services described in general in the *Cross Support Concept* document (reference [E3]) and in more detail in the *Cross Support Reference Model* (reference [1]).

c) Become familiar with basic Network Management concepts that are generally accepted in the industry. A useful treatment of this subject is contained in William Stallings' book, *The Practical Guide to Network Management*.

d) Finally, read this document in its given order. Begin with the introductory material and then consider the specific SLE management concepts presented in section 3. Then read the detailed descriptions of SLE management operations and SLE managed objects in sections 4 and 5.

e) Refer to the annexes as appropriate to understand some detailed aspects of the descriptive material contained in the body of the document.

1.7 DOCUMENT STRUCTURE

1.7.1 ORGANIZATION

This draft Recommendation is organized as follows:

a) Section 1 provides purpose, scope, applicability, and rationale of this draft Recommendation and identifies the conventions and references used throughout the document. This section also describes how the document is organized. A brief description is provided for each section and annex so that the reader will have an idea of where information can be found in the document. It also identifies terminology that is used in this document but defined elsewhere.

b) Section 2 provides an overview of SLE Service Management, places it in its context, and introduces some basic managed object concepts.

c) Section 3 establishes the operational aspects of service management. It defines the specific operations that are used to create, use, and delete instances of the managed object classes defined later in the document. These managed object instances are used to request the SLE services that a space mission needs, as well as to provide the information the SLE Complex needs to provide those services.

e) Section 4 defines the general characteristics of the managed objects that are exposed at the interface between the SLE Complex Management and the SLE Utilization Management.

f) Sections 5 through 14 specify each managed object used for SLE Service Management.

   1) Section 5 specifies common and abstract managed object classes.

   2) Section 6 specifies managed objects associated with the SLE data channel tree.

   3) Sections 7 through 14 specify managed objects used for transfer service production, storage, provision and use. The managed objects are arranged by their associated functional group (refer to the *Cross Support Reference Model*, reference [1], for a discussion of functional groups).

NOTE – A small number of SLE managed objects are specified in ancillary CCSDS Recommendations. These managed objects are identified in this document, along with references to the appropriate ancillary documents.

g) Annex A contains a list of acronyms.

h) Annex B provides a glossary of terms from the *Cross Support Reference Model* document (reference [1]) and from this SLE Service Management document.

i) Annex C specifies the rules for naming SLE services and managed objects.

j) Annex D illustrates the containment relationships between SLE managed objects.

k) Annex E lists informative references.

1.7.2 SLE SERVICES DOCUMENTATION TREE

1.7.2.1 This draft Recommendation is based on the cross support view of the architectural model defined in the *Cross Support Reference Model* (reference [1]). This document expands upon the concept of SLE Service Management as the interaction between SLE Utilization Management and SLE Complex Management for the purpose of defining, monitoring, and controlling SLE service packages.

This draft Recommendation is part of a suite of documents specifying the SLE services. The SLE document suite constitutes only one of three types of cross support services:

a) Part 1: SLE Services;†

b) Part 2: Ground Communications Services;

c) Part 3: Ground Domain Services.

The basic organization of the SLE Documentation is shown in figure 1-2.

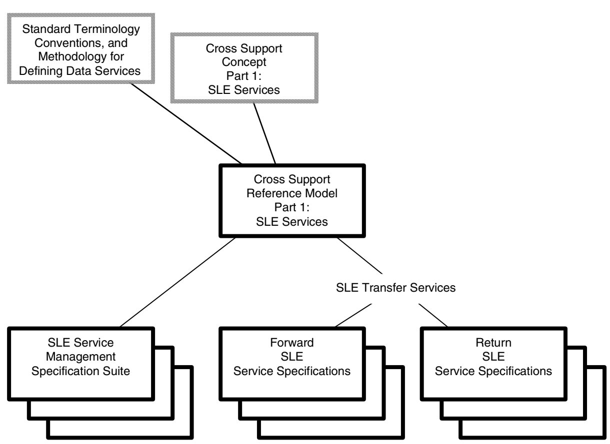


Figure 1-2: SLE Services Documentation

1.7.2.2 The specific SLE documents that are directly associated with this draft Recommendation are as follows:

†CCSDS Panel 3 is, at the time of publication, working only on Part 1 for SLE services.

a) *Standard Terminology, Conventions and Methodology (TCM) for Defining Data Services*. A Report identifying the existing international standards for defining SLE data services (reference [11]).

b) *Cross Support Concept—Part 1: Space Link Extension Services*. A Report introducing the concepts of cross support and SLE services (reference [E3]).

c) *Cross Support Reference Model—Part 1: Space Link Extension Services*. A Recommendation that defines the reference model, which provides a common basis for coordinating the development of CCSDS cross support Recommendations (reference [1]).

d) *SLE Forward Transfer Service Specifications*. Recommendations that will provide specification of all forward link SLE services (reference [E10]).

e) *SLE Return Transfer Service Specifications*. Recommendations that will provide specification of all return link SLE services (reference [E10]).

f) *SLE Service Management Specification*. The central document that establishes the SLE service management framework. Other documents included in the Service management suite include the following:

   1) *SLE Managed Object Formal Specification*. Recommendation(s) that will provide the formal specification of SLE managed objects using GDMO description techniques (reference [10]).

   2) *SLE Service Management Authentication for SLE Services*. Recommendation that will provide information pertaining to security for transfer and management services (reference [12]).

   3) *SLE—Service Management—Space Link Physical Layer Characteristics Managed Object Specification*. Recommendation(s) that will provide contents and structure for the Radio Frequency Characteristics information file (reference [13]).

   4) *SLE Service Management Implementation Mapping Rules*. A Recommendation that will describe how to map SLE management operations onto their primitives (reference [9]).

1.8 DEFINITIONS

1.8.1 DEFINITIONS FROM ASDC

This draft Recommendation uses the ASDC methodology, which provides the means to create a model of a system composed of objects that provide services to one another. The description of SLE Service Management uses abstract objects, which have abstract ports through which they provide abstract-services to other abstract objects. The following ASDC terminology is used in this document:

a) Abstract Object;

b) Abstract Service;

c) Port Types;

d) Service User/Provider;

e) Abstract Binding;

f) Abstract Operation;

g) Containment.

These terms are defined in reference [11].

1.8.2 DEFINITIONS FROM SLE SYSTEM ENVIRONMENT

The *Cross Support Reference Model* (reference [1]) defines the SLE system environment, data, and services. It introduces terms and concepts in the SLE system. In order to understand the draft *SLE—Service Management Specification*, the user must first understand the following terms:

a) Space Mission Data System;

b) Space Element;

c) Space Link;

d) Ground Element;

e) Space Link Extension System;

f) Mission Data Operation System (MDOS);

g) Mission User Entities;

h) SLE Utilization Management;

i) Space Link Data Unit;

j) Return Space Link Data Unit;

k) Forward Space Link Data Unit;

l) Space Link Data Channel;

m) Space Link Extension Service Data Unit;

n) Space Link Extension Data Channel;

o) SLE Transfer Service Instance;

p) SLE Service Package.

These terms are defined in reference [1].

1.8.3 DEFINITIONS FROM TELECOMMAND

This draft Recommendation uses the term ‘Mulitplexer Access Point’, which is defined in the Telecommand specifications (references [6] through [8]).

1.8.4 DEFINITIONS FROM OSI

This draft Recommendation uses the following terms and concepts defined in *OSI—The Directory: Models* (reference [E6]):

a) Action-parameter-pairs;

b) Attribute-value-pairs;

c) Distinguished Name (DN);

d) Relative Distinguished Name (RDN).

1.8.5 NOMENCLATURE

The following conventions apply throughout this draft Recommendation:

a) the words ‘shall’ and ‘must’ imply a binding and verifiable specification;

b) the word ‘should’ implies an optional, but desirable, specification;

c) the word ‘may’ implies an optional specification;

d) the words ‘is,’ ‘are,’ and ‘will’ imply statements of fact.

1.8.6 CONVENTIONS

Style and format conventions used to prepare this document are identified in the *Cross Support Reference Model* (reference [1]). Drawing conventions are defined for representing types of (abstract) objects and various instances of an object. Definitions are also provided for SLE Ports, Number of Ports for SLE services, and SLE Object Refinement.

In the containment diagrams, a line with a circle and diamond shape at each end indicates a containment relationship. The managed object with the diamond is contained by the managed object with the circle.

This draft Recommendation uses the typographic conventions contained in table 1-1.

Table 1-1: Typographic Conventions

FORMAT	MEANING
Bold	Names for attributes, constituents, actions, parameters, states, or notifications (definition).
Italic	Subsequent references to attributes, constituents, actions, parameters, states, or notifications.
'Single quotes'	Values for attributes, constituents, or parameters.
ALL CAPS	Operations.
<u>underline</u>	Emphasis only.

Usually it is clear from context whether 'managed object' refers to a managed object class, or to only one instance of that class. Wherever the meaning could be ambiguous, the appropriate term, class or instance, is used. By convention (reference [11]), the initial letter of the name of an object class is upper case, and the initial letter of the name of an instance is lower case. (e.g., a 'dataStore' managed object is an instance of the 'DataStore' managed object class).

1.9 REFERENCES

The following documents contain provisions that, through reference in this text, constitute provisions of this draft Recommendation. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this draft Recommendation are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS Recommendations.

[1] *Cross Support Reference Model—Part 1: Space Link Extension Services*. Recommendation for Space Data System Standards, CCSDS 910.4-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, May 1996.

[2] *Packet Telemetry*. Recommendation for Space Data System Standards, CCSDS 102.0-B-4. Blue Book. Issue 4. Washington, D.C.: CCSDS, November 1995.

[3] *Packet Telemetry Services*. Recommendation for Space Data System Standards, CCSDS 103.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, May 1996.

[4] *Telemetry Channel Coding*. Recommendation for Space Data System Standards, CCSDS 101.0-B-4. Blue Book. Issue 4. Washington, D.C.: CCSDS, May 1999.

[5] *Advanced Orbiting Systems, Networks and Data Links: Architectural Specification*. Recommendation for Space Data Systems Standards, CCSDS 701.0-B-2. Blue Book. Issue 2. Washington, D.C.: CCSDS, November 1992.

[6] *Telecommand Part 1—Channel Service*. Recommendation for Space Data System Standards, CCSDS 201.0-B-3. Blue Book. Issue 3. Washington, D.C.: CCSDS, June 2000.

[7] *Telecommand Part 2.1—Command Operation Procedures*. Recommendation for Space Data System Standards, CCSDS 202.1-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, October 1991.

[8] *Telecommand Part 3—Data Management Service*. Recommendation for Space Data System Standards, CCSDS 203.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, January 1987.

[9] *Space Link Extension—Service Management—Implementation Mapping Rules*. CCSDS Technical Panel Draft, CCSDS 910.12-W. White Book. n.p., n.d.

[10] *Space Link Extension—Service Management—Managed Object Formal Specification*. CCSDS Technical Panel Draft, CCSDS 910.6-W. White Book. n.p., n.d.

[11] *Standard Terminology, Conventions, and Methodology (TCM) for Defining Data Services*. Report Concerning Space Data Systems Standards, CCSDS 910.2-G-1. Green Book. Issue 1. Washington, D.C.: CCSDS, November 1994.

[12] *Space Link Extension—Service Management—Authentication for SLE Services*. CCSDS Technical Panel Draft, CCSDS 910.9-W. White Book. n.p., n.d.

[13] *Space Link Extension—Service Management—Space Link Physical Layer Managed Object Specification*. CCSDS Technical Panel Draft, CCSDS 910.7-W. White Book. n.p., n.d.

[14] K. Sollins and L. Masinter. *Functional Requirements for Uniform Resource Names.* December 1994. [RFC 1737]†

† Internet Request for Comments (RFC) texts are available on line in various locations (e.g., http://ietf.org/rfc/).

2 OVERVIEW

2.1 SLE SERVICE MANAGEMENT CONCEPTS

2.1.1 RELATION TO SLE TRANSFER SERVICES

SLE services comprise:

a) SLE transfer services (specified in reference [E10]), which transfer SLE data channels between the Space Element and the MDOS through the SLE system;

b) SLE service management (this draft Recommendation), which controls the scheduling and provision of SLE transfer service instances by the SLE system.

2.1.2 SLE SERVICE MANAGEMENT FUNDAMENTALS

2.1.2.1 SLE Service Management Environment

Figure 2-1 depicts the SLE architecture model defined in reference [10]. This model is defined according to the ASDC, as summarized in reference [9]. This draft Recommendation focuses on the interface indicated by the bold dashed line near the top of the figure. SLE Service Management, which is defined from the perspective of SLE Utilization Management, is the interaction between SLE Utilization Management and SLE Complex Management.

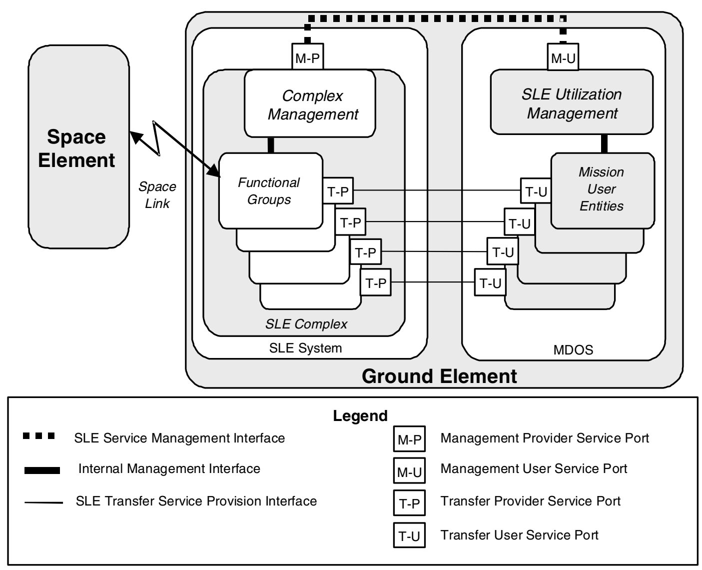


Figure 2-1: SLE Architecture

2.1.2.2 SLE Utilization Management

SLE Utilization Management is responsible for ensuring adequate telecommunication capabilities between the MDOS and the SLE Complexes that support the mission. The means by which this is done are outside the scope of this draft Recommendation.

2.1.2.3 SLE Complex Management

SLE Complex Management controls the extent to which SLE Utilization Management can affect actual SLE Complex resources. Because SLE Complex Management acts as an agent for SLE Utilization Management, only those aspects of the resources of an SLE Complex that SLE Complex Management chooses to expose are visible to SLE Utilization Management for management operations.

2.1.2.4 SLE Service Agreements

An SLE Service Agreement (referred to as a 'Service Agreement') holds information that enables management interaction between SLE Utilization Management and SLE Complex Management. This information is negotiated between the two entities by a means outside the scope of this draft Recommendation.

A serviceAgreement managed object represents a Service Agreement, and serves as a 'container' object for other SLE managed objects, including those that represent SLE service packages. Typically, one serviceAgreement managed object is created for a space mission, well before launch.

An MDOS and a Complex may engage in further bilateral agreements. In particular, they may negotiate service management policies and procedures for activities not covered by this draft Recommendation (e.g., active support for conflict resolution, accounting, billing, spacecraft emergency operations, etc.).

2.1.2.5 SLE Service Packages

An SLE service package (referred to as a 'service package') holds information about the types and provision periods for SLE transfer services, the end users that are allowed to access the services, and the desired production process configuration.

A servicePackage managed object represents a service package. Over the lifetime of a mission, many servicePackage managed objects may be created—one for each Space Link session. Additional servicePackage managed objects may be created to define offline services spanning multiple Space Link Sessions. Offline services may also be defined as part of a servicePackage managed object that defines online services.

2.1.2.6 SLE Service Package Time Span

The SLE service package lifetime is divided into two phases: the definition phase and the utilization phase.

In the definition phase, SLE Utilization Management formulates a service package according to the mission operations system's requirements by creating instances of managed object classes that carry mission-specific information for space link acquisition, service production and service delivery.

In the utilization phase, SLE Utilization Management may operate on these managed objects: inspect the current value of an attribute, set an attribute value if its behavior allows setting, invoke an action, and receive notifications.

2.2 SLE MANAGED OBJECT CONCEPTS

2.2.1 GENERAL

This draft Recommendation defines SLE Service Management in terms of managed objects and uses the concepts of containment and inheritance commonly used in object-oriented analysis and design (reference [E9]).

2.2.2 MANAGED OBJECTS

2.2.2.1 Service Management Requirements

This draft Recommendation presents a collection of managed object classes designed to satisfy a broad range of service management requirements in the following areas:

a) contracting between SLE Utilization Management and SLE Complex Management;

b) resource allocation planning within an SLE Complex and confirmation of commitment ahead of the actual service operations;

c) monitoring the production and provision processes by SLE Utilization Management and related reporting;

d) control of the production process during operation, directly by SLE Utilization Management, and automatically by a proxy when fast reaction to Complex-internal events is essential.

Each SLE managed object relates to an SLE Complex resource that contributes to the production and delivery of SLE transfer services.

2.2.2.2 Class vs. Instance

A managed object class is a template for managed object instances that have the same characteristics and behavior. A managed object instance is one specific occurrence of a given managed object class. One class may have many instances, but each instance is of only one class.

2.2.2.3 Managed Object Attributes

An attribute represents a characteristic of the resource represented by the managed object. Attribute values are used to configure, monitor or control the resource.

For any given mission, some attribute values are determined at design time, some during ground testing and calibration, and some during mission operations. Some may have a nominal value, which will remain fixed if the mission progresses normally, but which can be altered if contingencies arise. Many values are limited to a set of discrete values or to a range of values. Such limits may arise from spacecraft design, operational considerations, or the capabilities provided by an SLE Complex to the mission. Individual missions and SLE Complexes may establish implementation-unique constraints only when configuration parameter values are established.

2.2.2.4 Managed Object Behavior

SLE managed objects may exist in and transition between several **states**. In each state, certain behaviors are allowed and certain behaviors are disallowed.

SLE managed objects may perform **actions**, which are internal activities other than a change to an attribute value. For each state of each managed object, actions either succeed, fail, or are ignored.

Notifications allow SLE Utilization Management to trace events of interest, including state transitions, changing conditions, or other events. A managed object can issue a notification at any time, even if there is no telecommunication association between SLE Complex Management and SLE Utilization Management. Notifications that are issued when there is no telecommunication association are logged for later retrieval.

2.2.3 MANAGED OBJECT CONTAINMENT

Containment relationships between managed object classes reflect the dependencies between production steps and provision ports prescribed by the *Cross Support Reference Model* (reference [10]). Containment provides the flexibility to construct complex managed objects as needed to represent complex cross support requirements, but is also allows simpler managed objects to be constructed to represent less complex requirements. The containment rules provided by the managed object specifications in this draft Recommendation enable SLE Utilization Management to model the specific services needed for mission operations, as well as the resources required to provide those services.

Containment is also the basis for naming instances of SLE managed objects (see annex C).

2.2.4 MANAGED OBJECT INHERITANCE

Inheritance relationships between managed object classes are used to combine behavioral and structural elements shared by several classes. A derived class inherits the structure and behavior of its parent class.

2.2.5 MANAGED OBJECT MANIPULATION: OPERATIONS

SLE Utilization Management and SLE Complex Management interact by invoking and performing management operations on SLE managed objects.

A managed object can be created and deleted by a CREATE and a DELETE operation respectively. The value of an attribute can be inspected by a GET operation, and the value can be modified by a SET operation; at creation time it may have an initial value. An action is invoked by an ACTION operation with parameters. A managed object may issue notifications (using the NOTIFY operation), which signal to the outside world that a specific condition has occurred in the resource associated with the managed object.

3 SLE MANAGEMENT OPERATIONS AND BINDING

3.1 GENERAL

This section describes the establishment of a communication channel, and operational interaction via such a channel, between SLE Utilization Management and SLE Complex Management. For this purpose, this section defines terminology to help describe operational interaction and binding (3.3).

With respect to binding issues, this section:

a) outlines the binding process (3.3.2);

b) describes channel control mechanisms (3.3.3);

c) identifies the common knowledge that is prerequisite for any operational interaction (3.3.4);

d) describes binding-related requests (3.3.5);

e) identifies functional requirements for the support of binding (3.3.6).

With respect to operational interactions, this section:

a) presents the interaction model (3.4.1);

b) specifies the information contents of operation service primitives (3.4.2);

c) identifies functional requirements on the supporting infrastructure (3.3.6).

3.2 DEFINITIONS

For purposes of this Recommendation, the terms listed in this subsection are defined as follows:

a) Application (two types):

   1) Utilization Management (UM)-application denotes that part of a (possibly more comprehensive) SLE Utilization Management application dealing with the services to be provided by a single SLE Complex. This part is comprised of a piece of program together with the supporting part of the local infrastructure.

   2) Complex Management (CM)-application denotes that part of a (possibly more comprehensive) SLE Complex Management application dealing with the service provision for a single SLE Utilization Management. This part is comprised of: a piece of program that implements a single serviceAgreement managed object; all managed objects directly and indirectly contained in it (this collection of objects is referred to as a target-group); and the supporting components of the local infrastructure.

b) **Authority** (two types):

   1) **CM-authority** denotes the CM-application's administration authority.

   2) **UM-authority** denotes the UM-application's administration authority.

c) **Binding**:

   1) the collection of all information that invoker and performer must share to enable their cooperation;

   2) the process of establishing this information sharing;

   3) the establishment of a channel on the basis of the shared information.

d) **Channel**:

   1) the local infrastructure of a given invoker;

   2) the local infrastructure of a given performer;

   3) a communication network accessible to both infrastructures.

e) **Infrastructure**. All software components of a computer system that locally support a distributed application.

NOTE – Infrastructure includes operating system, telecommunication protocols, middleware and potential extensions such as middleware services or custom programmed add-ons, and APIs.

f) **Invoker**. All software components of a computer system that are involved in the issuing of invocation service primitives and in the processing of result and error returns (e.g., an SLE Utilization Management application).

g) **Local**. Denotes that something is pertinent to a single node.

h) **Managed Object**. An object-oriented concept particularly suited to capture network and system management capabilities.

i) **Marshalling**. The process of gathering data from one or more applications or non-contiguous sources in computer storage, putting the data pieces into a message buffer, and organizing or converting the data into a format that is prescribed for a particular receiver or programming interface.

NOTE – Marshalling is usually required when passing the output parameters of a program written in one language as input to a program written in another language.

j) **Node**. A computer system with network access that can support a channel involving another geographically remote node.

k) **Performer**. All local infrastructure and the supported object implementations for a specific application (e.g., .an SLE CM-application).

l) **Protocol**. The special set of rules for communicating used by the end points in a telecommunication connection when they send signals back and forth. Both end points must recognize and observe the protocol.

NOTES

1 Protocols are often described in an industry or international standard.

2 Protocols exist at several levels in a telecommunication connection. For instance, there are hardware telephone protocols, and there are also protocols between communicating programs within the same computer or at different locations.

m) **Service Primitive**. A conceptual representation of the information that passes between application and a channel. If the service primitive passes from the application to the channel, it is submitted; if it passes from the channel to the application, it is delivered.

n) **Target Group**. A collection of objects in the realm of the performer that may be destinations for operation invocations issued by the same invoker.

NOTE – For instance, a serviceAgreement instance and all its contained managed objects form the target group for a given UM-application.

o) **Target Object**. An object instance in the realm of the performer that is the final destination of an operation invocation.

NOTE – For instance, a servicePackage instance is a target object for a DELETE operation.

p) **Technology**. The Commercial Off-the-Shelf (COTS) items, supporting runtime environments, programming tools and formalisms employed in a specific infrastructure.

q) **Representation**. The technology-specific form of a piece of information.

NOTES

1 For SLE Service Management, the structure of a piece of information is formally specified in a technology independent way in the *Managed Object Formal Specification* (reference [10]).

2 The rules for the production of its corresponding technology-specific form are specified in *Implementation Mapping Rules* (reference [9]).

3.3 BINDING

3.3.1 GENERAL

Operation interaction can occur only when the local parts of a channel are configured consistently. Binding ensures that this happens.

This subsection:

a) outlines the principles of the binding process in 3.3.2;

b) identifies the information to be shared in 3.3.3;

c) specifies information items in binding-related requests in 3.3.4;

d) identifies functional capabilities needed to support binding in 3.3.5.

3.3.2 BINDING PROCESS

Before operational interaction can occur, the interaction partners—the UM-application and the corresponding SLE Complex Management—must exchange enabling information and accordingly configure their respective local channel infrastructures. That exchange shall follow the following process:

Step 1) A CM-application’s administration authority publishes the binding information necessary for establishing a channel to it.

Step 2) A UM-application’s administration authority retrieves the published binding information.

Step 3) The CM-application’s administration authority configures the local part of the channel.

Step 4) The UM-application requests its local infrastructure to establish a channel to the CM-application.

Step 5) The UM-application’s administration authority configures the local part of the channel according to the binding information published by the CM-authority.

Step 6) The UM-application’s local infrastructure establishes a telecommunication association with the CM-application’s local infrastructure.

Step 7) Both local infrastructures cooperate to establish a channel.

Step 8) The UM-application’s local infrastructure notifies the UM-application that a channel is established.

NOTE – After this step, the UM-application and the CM-application are said to be 'bound'.

Step 9) UM- and CM-applications cooperate as described in 3.4.1.

NOTES

1 Steps (1) and (2) of the process require negotiations between the UM- and CM-authorities. The information to be agreed is identified in 3.3.4.

2 Steps (3) and (4) must be performed by the respective local system management function.

3 The local interaction between a UM-application and its supporting local infrastructure in steps (5) and (8) is for further study.

4 Step (6) requires information to be exchanged between the local infrastructures of the UM- and CM-applications. These information items are covered in 3.3.5.

5 An approach for port relocation is for further study.

3.3.3 CHANNEL CONTROL

3.3.3.1 General

In addition to actual binding, a channel may be suspended and resumed and, when no further interaction is required, the applications can unbind and release the channel resources.

3.3.3.2 Suspension and Resumption

Step 10) The UM-application requests its local infrastructure to suspend the channel and thus causes the local infrastructure to send a CHANNEL-SUSPEND request to the CM-application's local infrastructure. After receipt of a corresponding CHANNEL-SUSPEND result, it terminates the telecommunication association. The channel configurations and the resources allocated to it in both local infrastructures are preserved.

Step 11) The UM-application requests its local infrastructure to resume the channel and thus causes the local infrastructure to re-establish the telecommunication association and send a CHANNEL-RESUME request to the CM-application's local infrastructure. After receipt of a corresponding CHANNEL-RESUME result, all channel resources are enabled for normal operation.

Step 12) UM- and CM-applications cooperate as described in 3.4.1.

NOTES

1 The local interaction between a UM-application and its supporting local infrastructure in steps (10) and 11) is for further study.

2 Steps (10) and (11) require information to be exchanged between the local infrastructures of the UM- and CM-applications. These information items are covered in 3.3.4 and 3.3.5.

3.3.3.3 Explicit Unbinding

Step 13) The UM-application requests its local infrastructure to destroy the channel and thus causes the local infrastructure to send an UNBIND request to the CM-application's local infrastructure. After receipt of a corresponding UNBIND result, it terminates the telecommunication association. The channel configurations and the resources allocated to it in both local infrastructures are released.

Step 14) At any time after step (8), the UM-application or the CM-application may request its local infrastructure to abort the channel and thus cause the local infrastructure to send a PEER-ABORT request to the cooperating infrastructure immediately before the channel's local resources are released.

Step 15) If the channel includes a gateway and this gateway detects a malfunction of a local infrastructure, it sends a PEER-ABORT request to the other local infrastructure and releases its channel-dedicated resources immediately thereafter.

NOTES

1 The local interaction between a UM-application and its supporting local infrastructure in steps (13) and (14) is for further study.

2 Steps (13), (14), and (15) require information to be exchanged between the local infrastructures of the UM- and CM-applications. These information items are covered in 3.3.4 and 3.3.5.

3.3.3.4 Implicit Unbinding

Different technologies may handle a breakdown of the underlying telecommunication association differently; the breakdown may have the effect of an UNBIND request, of a CHANNEL-SUSPEND request or of a PEER-ABORT request. The real effect must be specified in the applicable technology mapping rules.

3.3.4 IDENTIFICATION OF BINDING INFORMATION

a) In order to support the binding process, the authorities responsible for cooperating applications must agree to adhere to the following pieces of binding information for implementation, configuration and operation of a channel:

   1) a specification of a channel-identifier, which refers to the collection of binding information as identified below;

   2) a UM-identifier for the UM-application;

   3) a CM-identifier for the CM-application.

b) With regard to the network component of a channel, the following information must be agreed upon:

   1) identification of a network that provides the necessary connectivity;

   2) identification of the node coordinates of the computer system that hosts the UM-applications;

   3) identification of the node coordinates of the computer system that hosts the CM-applications;

   4) identification of an access point for network management requests.

c) With regard to a channel's telecommunication components, the following information must be agreed upon:

   1) identification of the telecommunication protocol to be used;

   2) specification of the configuration of the telecommunication protocol;

   3) specification of a return time-out period, which defines the maximum time between submission of an invocation primitive to the channel and delivery of the related result or error primitive;

   4) identification of the technology to be used to implement the local parts of the channel.

NOTE – If the local parts are implemented with different technologies, then the following must be agreed upon: identification of a technology gateway, including a specification of its node coordinates; identification of the gateway configuration; assignment of the responsibility for gateway configuration and management; and identification of an access point for gateway management requests.

d) With regard to security measures to be applied for the cooperation between SLE Utilization Management and SLE Complex Management via a channel, the following information must be agreed upon:

1) identification of the security measures to be applied;

2) specification of the respective security information.

e) With regard to a channel's general support for managed objects, the following information must be agreed upon:

1) Identification of the applicable suite of CCSDS Recommendations, which may include *Service Management* (this document), *Managed Object Formal Specification* (reference [10]), *Implementation Mapping Rules* (reference [9]), and *Authentication* (reference [12]);

2) identification of the capability set of managed objects supported by the target Complex.

f) With regard to the direct interaction between channel and managed objects, the identification of one or more *agreed* serviceAgreement managed objects, which identifies a target group, must be agreed upon.

g) With regard to the direct support of specific managed objects, the following information must be agreed upon:

1) for an antennaPointing managed object, identification of the protocol to be used for the transfer of trajectory files;

2) for a antennaPointing managed object, specification of a location for the deposit and retrieval of trajectory files;

3) for a antennaPointing managed object, a presentation format for the trajectory information contained in a trajectory file;

4) for an eventHandler managed object, specification of the list of events to be supported by the target Complex.

NOTES

1 Information items described in subsection 3.3.4 that are carried in channel-related requests need a representation compatible with the technology by which the requests are implemented. Such representations must be specified in the respective technology mapping rules. Examples for such items are found in 3.3.4(a), (1) through (3), as well as 3.3.4(c)(5) and 3.3.4(g)(1).

2 UM- and CM-authorities may negotiate additional items that have no direct impact on establishment and use of channels. Examples include: provision of services that are not covered by existing CCSDS Recommendations (e.g. ranging and range rate measurements, delivery of telemetry data on storage media), compensation for the management and transfer services provided, handling of spacecraft emergencies, conflict resolution schemes (e.g. mission priorities), and test arrangements with simulated spacecraft.

3 Future implementation of the SLE Service Management Interface may support dynamic negotiation of selected items from the binding information identified above.

3.3.5 INFORMATION IN CHANNEL-RELATED REQUESTS

3.3.5.1 BIND

3.3.5.1.1 General

a) **Purpose**. With a BIND request, an application informs its local infrastructure that establishment of a channel is required. Performing a BIND request includes:

1) establishing access to the telecommunication network identified in the binding information;

2) configuration of the network access protocol stack;

3) activation of the protocol stack.

NOTE – The implementation of a BIND request is highly technology-specific and relies on the correct interpretation and use of binding information by all parties involved.

b) **Invoker**. The BIND operation can be submitted by a UM-application.

c) **Performer**. A BIND request is performed by the local infrastructures of the nodes involved in the channel to be established.

3.3.5.1.2 BIND Request Information Items

The following information items of a BIND request shall be processed by the local infrastructure of the remote node:

a) **invoker-credentials.** Enables the performer of the operation to authenticate the invoker/requester and to verify its access rights.

1) If this parameter is set to ‘unused’, no security measures shall be applied.

2) Any other value is specific to a particular security technology and must be provided according to reference [12].

3) The invoker must provide this information item according to the applicable syntax.

4) The performer shall use the respective entry in the Security Information Base (SIB) for verification at runtime. The respective SIB entry shall be accessible via an *SLE-entity-identifier* in the target managed object.

b) **invoke-id.** Provides an identification of the invocation, which subsequent result- or error-primitives use to uniquely identify the original invocation.

1) The syntax for this information item shall be technology-specific.

2) The invoker must supply it according to reference [9].

c) **operation-type** (= 'BIND'). This information item identifies the type of operation to be performed.

1) The need for and representation of this information item shall be technology-specific.

2) If needed, the invoker must supply it according to the applicable recommendation on technology mapping rules.

d) **primitive-kind** (= 'request'). The value 'request' distinguishes this information structure from operations on Managed Objects (MO) (see 3.4.2 for the operations).

1) The need for and representation of this information item shall be technology-specific.

2) If needed, the invoker must supply it according to reference [9].

e) **channel-id.** This item identifies the channel to be created. It is used by the receiving infrastructure to verify the current configuration of the local channel infrastructure.

f) **service-agreement-dn.** The DN of the specific service agreement to which the UM-application is referring in this management association.

g) **UM-identifier.** This item identifies the UM-authority on whose behalf the channel is constructed (see 3.2). The UM-identifier permits the responding SLE management entity to determine whether the initiating management entity has access privileges, and which method to use to authenticate the *invoker-credentials*.

1) If the UM cannot be identified, the request shall be ignored; no information shall be returned to the initiating node.

2) Data type/values. The value shall be a visible string (i.e., alphanumeric characters, "-", "_", no white spaces, not case-sensitive).

3) The value shall be fixed commonly by SLE Complex Management and SLE Utilization Management when the service agreement is negotiated.

NOTE – The value need not become visible in the serviceAgreement managed object; however, it must be entered into the Authentication Information Base.

3.3.5.1.3 BIND Result Information Items

The following information items of a BIND request shall be returned by the remote node's local infrastructure to the requesting node's local infrastructure.

a) **performer-credentials.** This item allows SLE Utilization Management to authenticate the return.

b) **invoke-id.** See 3.3.5.1.2.

c) **primitive-kind** (= 'result'). See 3.3.5.1.2.

3.3.5.1.4 BIND Error Information Items

a) **performer-credentials.** See 3.3.5.1.3.

b) **invoke-id.** See 3.3.5.1.2.

c) **primitive-kind**. (= 'error') See 3.3.5.1.2.

d) **diagnostic.** This information item provides the reason the request was rejected, as follows:

   1) 'unable to comply'. The performer was unable to accept the request because of a fault affecting the infrastructure or application involved.

   2) 'invalid service agreement'. The requested service agreement does not exist or is not *applicable* at the time of the invocation.

   3) 'invoker unknown'. An SIB entry for the invoker could not be found.

   4) 'access denied'. A SIB entry for the invoker could be found; however its access rights are not sufficient for the invoked operations.

   5) 'other'. The request was rejected for reasons other than identified above.

3.3.5.2 UNBIND

3.3.5.2.1 General

a) **Purpose**. An UNBIND request destroys the channel specified. After performance of an UNBIND request, access to the telecommunication network is disabled, and all local channel resources are released.

b) **Invoker**. An UNBIND request can be submitted only by the SLE Utilization Management.

c) **Performer**. An UNBIND request is performed by the UM-application's local infrastructure in cooperation with the remote CM-application's local infrastructure.

3.3.5.2.2 UNBIND Request Information Items

a) **invoker-credentials**. See 3.3.5.1.2.

b) **invoke-id**. See 3.3.5.1.2.

c) **operation-type** (= 'UNBIND'). See 3.3.5.1.2.

d) **primitive-kind** (= 'request'). See 3.3.5.1.2.

e) **channel-id**. This item identifies the channel to be destroyed. The receiving infrastructure uses it to verify the local channel infrastructure's current configuration.

3.3.5.2.3 UNBIND Result Information Items

a) **performer-credentials**. See 3.3.5.1.3.

b) **invoke-id**. See 3.3.5.1.2.

c) **operation-type** (= 'UNBIND'). See 3.3.5.1.2.

d) **primitive-kind** (= 'result'). See 3.3.5.1.2.

e) **channel-id**. This item identifies the channel to be destroyed. The receiving infrastructure uses it to verify the local channel infrastructure's current configuration. The representation of this item is specified in 3.3.4.

3.3.5.2.4 UNBIND Error Information Items

a) **performer-credentials**. See 3.3.5.1.3.

b) **invoke-id**. See 3.3.5.1.2.

c) **operation-type** (= 'UNBIND'). See 3.3.5.1.2.

d) **primitive-kind** (= 'error'). See 3.3.5.1.2.

e) **diagnostic**

   1) 'unable to comply'. See 3.3.5.1.4.

   2) 'invalid channel'. The channel to destroy does not exist.

   3) 'access denied'. See 3.3.5.1.4.

   4) 'other'. See 3.3.5.1.4.

3.3.5.3 PEER-ABORT

3.3.5.3.1 General

a) **Purpose**. The PEER-ABORT request causes the channel between SLE Complex Management and SLE Utilization Management to be aborted.

   1) The request allows the invoker to notify the peer system that the local application detected an error that requires the channel between them to be destroyed.

   2) The PEER-ABORT request is truly abortive. This implies that the performance of this invocation shall not be delayed by other operations invoked earlier and possibly being buffered by the underlying communication service. This must be considered when the mapping onto the underlying communication services is decided.

b) **Invoker**. This request may be invoked by SLE Complex Management or by a gateway (if included in the channel).

c) **Performer**. This request is performed by the local channel infrastructure of both SLE Complex Management and SLE Utilization Management.

3.3.5.3.2 PEER-ABORT Request Information Items

a) **invoker-credentials**. See 3.3.5.1.2.

b) **invoke-id**. See 3.3.5.1.2.

c) **operation-type** (= ‘PEER-ABORT’). See 3.3.5.1.2..

d) **primitive-kind** (= ‘request’). See 3.3.5.1.2..

e) **diagnostic**

   1) ‘unable to comply’. See 3.3.5.1.4.

   2) ‘operational requirement’. .The local system had to destroy the channel in order to accommodate other demands.

   3) ‘protocol error’. The local application detected an error in the sequencing of management operations.

   4) ‘communications failure’. The communications service on the other side of the gateway was disrupted.

   NOTE – This value only applies if the channel includes a gateway.

   5) ‘encoding error’. The local application detected an error in the encoding of a service primitive.

6) ‘return timeout’. The local application detected that an operation’s return primitive was not received within a specified period of time.

7) ‘unexpected invoke-id’. The local application received a return with an *invoke-id* that did not match the *invoke-id* of any operation for which a return was pending.

8) ‘end of service agreement period’. The local application detected that the service agreement period ended and SLE Utilization Management did not invoke an UNBIND operation.

9) ‘other’. See 3.3.5.1.4.

3.3.5.3.3 PEER-ABORT Result and Error Information Items

PEER-ABORT Result and Error Information Items are not supported, because the channel at the potential destination may already be partially destroyed.

3.3.5.4 CHANNEL-SUSPEND

3.3.5.4.1 General

a) **Purpose**. A CHANNEL-SUSPEND request disables access to the telecommunication network supporting the channel identified.

NOTES

1 Together with CHANNEL-RESUME, this feature may be used to reduce communication cost.

2 A CHANNEL-SUSPEND request does not affect the allocation and configuration of the local channel resources.

b) **Invoker**. Only SLE Utilization Management can submit a CHANNEL-SUSPEND request.

c) **Performer**. A CHANNEL-SUSPEND request is performed by the UM-application’s local infrastructure in cooperation with the remote CM-application’s local infrastructure.

3.3.5.4.2 CHANNEL-SUSPEND Request Information Items

a) **invoker-credentials**. See 3.3.5.1.2.

b) **invoke-id**. See 3.3.5.1.2.

c) **operation-type**. (= ‘CHANNEL-SUSPEND’) See 3.3.5.1.2.

d) **primitive-kind**. (= ‘request’) See 3.3.5.1.2.

e) **channel-id**. This item identifies the channel in which network access shall be disabled.. See 3.3.5.1.2.

3.3.5.4.3 CHANNEL-SUSPEND Result Information Items

a) **performer-credentials**. See 3.3.5.1.3.

b) **invoke-id**. See 3.3.5.1.2.

c) **operation-type**. (= 'CHANNEL-SUSPEND') See 3.3.5.1.2.

d) **primitive-kind**. (= 'result') See 3.3.5.1.2.

3.3.5.4.4 CHANNEL-SUSPEND Error Information Items

a) **performer-credentials**. See 3.3.5.1.3.

b) **invoke-id**. See 3.3.5.1.2.

c) **operation-type** (= 'CHANNEL-SUSPEND'). See 3.3.5.1.2.

d) **primitive-kind** (= 'error'). See 3.3.5.1.2.

e) **diagnostic**

   1) 'unable to comply'. See 3.3.5.1.2.

   2) 'invalid channel'. The channel to destroy does not exist.

   3) 'access denied'. See 3.3.5.1.2.

   4) 'other'. See 3.3.5.1.2.

3.3.5.5 CHANNEL-RESUME

3.3.5.5.1 General

a) **Purpose**. A CHANNEL-RESUME request re-establishes access to the telecommunication network supporting the identified channel.

NOTE – This feature may be used together with a CHANNEL-SUSPEND request to reduce communication cost.

b) **Invoker**. Only SLE Utilization Management can submit a CHANNEL-RESUME request.

c) **Performer**. A CHANNEL-RESUME request is performed by the UM-application's local infrastructure in cooperation with the remote CM-application's local infrastructure.

3.3.5.5.2 CHANNEL-RESUME Request Information Items

a) **invoker-credentials**. See 3.3.5.1.2.

b) **invoke-id**. See 3.3.5.1.2.

c) **operation-type** (= 'CHANNEL-RESUME'). See 3.3.5.1.2.

d) **primitive-kind** (= 'request'). See 3.3.5.1.2.

e) **channel-id**. This item identifies the channel in which network access shall be re-enabled. See 3.3.5.1.2. The receiving infrastructure uses it to verify the local channel infrastructure's current configuration.

3.3.5.5.3 CHANNEL-RESUME Result Information Items

a) **performer-credentials**. See 3.3.5.1.3.

b) **invoke-id**. See 3.3.5.1.2.

c) **operation-type** (= 'CHANNEL-RESUME'). See 3.3.5.1.2.

d) **primitive-kind** (= 'result'). See 3.3.5.1.2.

e) **channel-id**. This item identifies the channel in which network access shall be re-enabled. See 3.3.5.1.2. The receiving infrastructure uses it to verify the local channel infrastructure's current configuration.

3.3.5.5.4 CHANNEL-RESUME Error Information Items

a) **performer-credentials**. See 3.3.5.1.3.

b) **invoke-id**. See 3.3.5.1.2.

c) **operation-type** (= 'CHANNEL-RESUME'). See 3.3.5.1.2.

d) **primitive-kind** (= 'error'). See 3.3.5.1.2.

e) **diagnostic**

   1) 'unable to comply'. See 3.3.5.1.4.

   2) 'invalid channel'. The channel to resume does not exist.

   3) 'access denied'. See 3.3.5.1.4.

   4) 'other'. See 3.3.5.1.4.

3.3.6 FUNCTIONAL REQUIREMENTS TO SUPPORT BINDING

3.3.6.1 General

Binding requires support by information repositories and a minimal support by infrastructure functions for processing channel-related requests.

3.3.6.2 Supporting Information Repositories

To support the binding process, the following information repositories must exist and be accessible to all parties involved in the binding process (see 3.3.2):

a) a repository for type definitions of all data types that may occur in binding requests or in service primitives;

NOTE – Currently, CCSDS does not plan to establish a common repository for CCSDS-approved type definitions for access by member Agencies. The *Managed Object Formal Specification* (reference [10]) and the *Implementation Mapping Rules* (reference [9]), together with a rigorous programming discipline, are deemed to supply type information that is sufficient to ensure consistent support for managed object operations by all UM- and CM-applications.

b) a repository for security information;

NOTE – The conceptual contents and operations of the Security Information Base are specified in reference [12].

c) a repository for telecommunication protocol configuration information;

NOTE – The conceptual contents and operations on a Telecommunication Information Base are specified in 5.19.

d) an *agreed* serviceAgreement managed object.

NOTE – The conceptual contents and operations on serviceAgreement managed objects is described in this draft Recommendation in 5.14.

3.3.6.3 Supporting Infrastructure Functions

In order to support the processing of channel-related requests and to enable the desired channel behavior, the local infrastructure of a performer must be able to:

a) access the identified telecommunication network;

b) establish, maintain and release a telecommunication association;

c) identify channel-related request primitives;

d) construct, encode and decode channel-related request primitives;

e) resolve names found in channel-related request primitives;

f) construct the local part of a channel according to the channel type referenced by a *channel-id*;

g) access the information repositories identified in (a) through (f) above and configure the local part of the channel, constructed according to the applicable information found there;

h) preserve the resources and configuration of the local part of a channel, even if the telecommunication association is terminated;

i) release the resources of the local part of a channel;

j) associate the serviceAgreement implementation with the channel constructed via a stub suitable for marshalling.

NOTE – Typically, state-of-the-art COTS middleware provides a large range of functions and services, which presumably cover most of the capabilities identified in (a) through (j) above. Nevertheless, it may be possible that a capability is not available off-the-shelf. In this case, a trade-off shall decide whether or not:

- the demanded capability shall be implemented as an add-on (to the technology or to the application);
- the capability requirement shall be relaxed;
- the capability requirement shall be dropped.

The mapping rules for the respective technology must reflect the trade-off made and the decision.

3.4 OPERATIONAL INTERACTION BETWEEN SLE UTILIZATION MANAGEMENT AND SLE COMPLEX MANAGEMENT

3.4.1 OPERATIONAL INTERACTION MODEL

After exchange of the respective binding information between administrations for applications A and B, and after the appropriate configuration of the local parts of a prospective channel, operational interaction is enabled. Binding process and binding information are described in 3.3.

Operational interaction shall follow the following pattern:

Step 1) Application A composes an invocation service primitive for some operation on a target object from the target group and submits it to the channel for transmission. Application A is the invoker for this operation.

Step 2) The performer's local infrastructure analyzes this service primitive. If the infrastructure discovers a problem, it forms an error primitive with an appropriate diagnostic, and uses the channel for its transmission back to the invoker. Otherwise, it activates the target object and delivers the invocation to it.

Step 3) The target object analyzes the invocation service primitive. If the target object discovers a problem, it forms an error primitive with an appropriate diagnostic and submits it to the channel for transmission back to the invoker. Otherwise, it performs the operation invoked and constructs a result service primitive, which it submits to the channel for transmission back to the invoker.

Step 4) After step 2, the invoker waits for the delivery of an error or return service primitive. Upon its delivery from the channel, it continues processing.

Step 5) Steps 1 through 4 are repeated as long as required by the invoker.

3.4.2 SPECIFICATION OF INFORMATION IN SERVICE PRIMITIVES

3.4.2.1 Common Information Items

3.4.2.1.1 General

This subsection specifies information items that are common to most service primitives.

3.4.2.1.2 Common Information Items of Invocation Service Primitives

Each invocation service primitive shall provide the following information items:

a) **invoker-credentials**. See 3.3.5.1.2.

b) **invoke-id**. See 3.3.5.1.2.

c) **target-mo-name.** This information item identifies the managed object that shall perform the operation (if the invocation passed all other stages of verification by the appropriate local infrastructure).

   1) The syntax for this information shall be technology-specific.

   2) The performer must provide a value according to the rules for the translation of Distinguished Names into the technology-specific name. These rules are specified in reference [9].

d) **operation-type.** This information item identifies the type of operation to be performed.

   1) This information item shall assume one of the following (symbolic) values: 'GET', 'SET', 'ACTION', 'NOTIFY', 'CREATE', 'DELETE', 'DISCOVER'.

2) The need for and representation of this information item shall be technology-specific.

3) If needed, the invoker must supply it according to the applicable recommendation on technology mapping rules.

e) **primitive-kind.** This information item identifies the kind of this primitive. See 3.3.5.1.2.

   1) For an invocation service primitive, it shall have the value 'invocation'.

   2) The need for and representation of this information item shall be technology-specific.

   3) If needed, the invoker must supply it according to reference [9]

f) Further information items are operation-specific and are described with the respective operation.

3.4.2.1.3 Common Information Items of Result Service Primitives

Each result service primitive shall provide the following information items:

a) **performer-credentials.** See 3.3.5.1.3.

b) **invoke-id.** See 3.3.5.1.2.

c) **primitive-kind.** For a result service primitive, it shall have the value 'result'. See 3.3.5.1.2 for more detail.

d) Further result information items are operation-specific and are described with the respective operation.

3.4.2.1.4 Common Information Items of Error Service Primitives

Each error service primitive shall provide the following information items:

a) **performer-credentials.** See 3.3.5.1.3.

b) **invoke-id.** See 3.3.5.1.2.

c) **primitive-kind.** For an error service primitive, this information item shall have the value 'error'. See 3.3.5.1.2 for more detail.

d) **diagnostic.** This information item provides a reason the operation invocation was rejected.

   1) As a result of an analysis of the invocation by the performer, this information item shall be set to one of the (symbolic) values specified below or in the correspondent subsections on specific operations.

i) ‘unable to comply’. The performer was unable to accept the invocation because of a fault affecting the infrastructure or application involved.

ii) ‘invoker unknown’. See 3.3.5.1.4.

iii) ‘access denied’. See 3.3.5.4.4.

iv) ‘target-mo unknown’. The target managed object does not exist or it is not visible to the UM-application.

v) ‘operation not supported’. The target managed object does not support the operation specified in the invocation.

vi) ‘other’. The operation failed for reasons other than those identified above.

2) Further diagnostic values are operation-specific and are described with the respective operation.

3) The representation of a value is technology-specific. The performer must supply it according to the applicable recommendation on technology mapping rules.

4) The applicable mapping rules may specify that certain diagnostics are not supported. In this case, the ‘other’ diagnostic shall be used.

3.4.2.2 GET

3.4.2.2.1 General

a) **Purpose**. The GET operation shall return the current values of selected target managed object’s attributes.

b) **Invoker**. This operation may be invoked by SLE Utilization Management or by an eventHandler managed object.

c) **Performer**. This operation shall be performed by the UM-application.

3.4.2.2.2 GET Invocation Primitive Information Items

a) **invoker-credentials**. See 3.3.5.1.2.

b) **invoke-id**. See 3.3.5.1.2.

c) **operation-type** (= ‘GET’). See 3.4.2.1.2.

d) **primitive-kind** (= ‘invocation’). See 3.3.5.1.2.

e) **target-mo-name**. This item identifies the managed object from which the current attribute values are acquired. For further details see 3.4.2.1.2.

f) A set of ‘**attribute-ids**’. An *attribute-id* specifies the managed object attribute whose value is to be acquired by the GET operation.

1) The representation of this information item shall be technology-specific. The invoker must supply *attribute-id* according to the attribute naming conventions of the technology employed.

2) Technology considerations may allow the support of not more than one *attribute-value-pair*. A respective constraint must be specified in the applicable technology mapping rules.

3.4.2.2.3 GET Result Primitive Information Items

a) **performer-credentials**. See 3.3.5.1.3.

b) **invoke-id**. See 3.3.5.1.2.

c) **primitive-kind** (= 'result'). See 3.3.5.1.2.

d) A set of '**attribute-value-pairs**'. An *attribute-value-pair* consists of an *attribute-id* paired with its *value* (*attribute-id*, *value*), where *attribute-id* is the name of an attribute defined in the managed object class definition and *value* is a value for this attribute taken from the data type and range defined for that attribute.

   1) The representation of this information item is technology-specific. The performer must supply

      i) *attribute-id* according to the attribute naming conventions of the technology employed.

      ii) *value* according to type/value representation of the technology employed.

   2) Technology considerations may allow the support of not more than one *attribute-value-pair*. A respective constraint must be specified in the applicable technology mapping rules.

3.4.2.2.4 GET Error Primitive Information Items

a) **performer-credentials**. See 3.3.5.1.3.

b) **invoke-id**. See 3.3.5.1.2.

c) **primitive kind** (= 'error'). See 3.3.5.1.2.

d) **diagnostic**

   1) 'unable to comply'. See 3.3.5.1.4

   2) 'invoker unknown'. See 3.4.2.1.4.

   3) 'access denied'. See 3.3.5.4.4.

   4) 'target-mo unknown'. See 3.4.2.1.4.

   5) 'operation not supported'. See 3.4.2.1.4.

6) ‘unknown attribute’. One (or more) *attribute-id* was not recognized by the subject managed object. The *diagnostic* contains the additional information: set of *attribute-id* of each unknown attribute

7) ‘other’. See 3.3.5.1.4.

3.4.2.3 SET

3.4.2.3.1 General

a) **Purpose**. The SET operation sets each of the target managed object’s listed attributes to the corresponding value specified in the list.

b) **Invoker**. This operation can be invoked by the UM-application or on its behalf by an eventHandler managed object.

c) **Performer**. This operation can be performed by a CM-application.

   1) A SET operation shall be performed in a sequence specific to the Complex.

   2) If any portion of the SET fails, the entire operation shall fail.

   NOTE – The effects of a failed SET operation on a Complex’s resources are undetermined.

3.4.2.3.2 SET Invocation Primitive Information Items

a) **invoker-credentials**. See 3.3.5.1.2.

b) **invoke-id**. See 3.3.5.1.2.

c) **operation-type** (=‘SET’). See 3.4.2.1.2.

d) **primitive-kind** (= ‘invocation’). See 3.3.5.1.2.

e) **target-mo-name**. This item specifies the managed object whose attributes shall be set. For further details, see 3.3.5.1.2.

f) A set of ‘**attribute-value-pairs**’. An *attribute-value-pair* consists of an *attribute-id* paired with its *value* (*attribute-id, value*), where *attribute-id* is the name of an attribute defined in the managed object class definition designated by *target-mo-name*, and *value* is a value for this attribute taken from the data type and range defined for that attribute.

   1) The setting of an attribute may be constrained to specific values. These constraints are identified in the description of the respective attribute. This draft Recommendation assumes that the performer can verify such constraints and can cause a SET operation to fail if the constraints are not met.

   2) The representation of this information item is technology-specific. The invoker must supply the *attribute-id* according to the attribute naming conventions of the

technology employed; and the attribute *value* according to type/value representation of the technology employed.

3.4.2.3.3 SET Result Primitive Information Items

a) **performer-credentials**. See 3.3.5.1.3.

b) **invoke-id**. See 3.3.5.1.2.

c) **primitive-kind** (= 'result'). See 3.3.5.1.2.

3.4.2.3.4 SET Error Primitive Information Items

a) **performer-credentials.** See 3.3.5.1.3.

b) **invoke-id.** See 3.3.5.1.2.

c) **primitive-kind.** (= 'error') See 3.3.5.1.2.

d) **diagnostic**

   1) 'unable to comply'. See 3.3.5.1.4.

   2) 'invoker unknown'. See 3.4.2.1.4.

   3) 'access denied'. See 3.3.5.4.4.

   4) 'target-mo unknown'. See 3.4.2.1.4.

   5) 'operation not supported'. See 3.4.2.1.4.

   6) 'invocation disabled in current state'. In the managed object's current state, the SET operation is not allowed.

   7) 'unknown attribute'. One or more *attribute-id(s)* was not recognized by the subject managed object. The *diagnostic* contains the additional information: set of *attribute-id* for each unknown attribute.

   8) 'invalid attribute value'. One (or more) *value(s)* did not match the data type of the respective attribute. The *diagnostic* contains the additional information: set of *attribute-id* for each unknown attribute.

   9) 'constraint violation'. One (or more) *attribute-value-pairs* violate a constraint for the values of that attribute.

   10) 'other'. See 3.3.5.1.4.

3.4.2.4 ACTION

3.4.2.4.1 General

a) **Purpose**. The ACTION operation causes the managed object to perform the specified action with the values specified for the various action parameters. ACTION operations are specific to managed objects and are described with them.

NOTES

1 During the definition phase, an ACTION operates on the Complex's planning resources; during the utilization phase, it operates on the Complex's operational resources.

2 A possible side effect of the performance of an ACTION is the change of an attribute value.

b) **Invoker**. An operation can be invoked by SLE Utilization Management or on its behalf by an eventHandler managed object.

c) **Performer**. An ACTION operation is performed by the CM-application.

3.4.2.4.2 ACTION Invocation Primitive Information Items

a) **invoker-credentials**. See 3.3.5.1.2.

b) **invoke-id**. See 3.3.5.1.2.

c) **operation-type** (= 'ACTION'). See 3.4.2.1.2.

d) **primitive-kind** (= 'invocation'). See 3.3.5.1.2.

e) **target-mo-name**. This item identifies the managed object that shall perform the ACTION operation. For further details see 3.3.5.1.2.

f) **action-id**. This is the name of the action to be performed.

   1) Valid *action-ids* are specified in the corresponding managed object class.

   2) The representation of this information item is technology-specific. The invoker must supply *action-id* according to the action naming conventions of the technology employed.

g) A set of **'action-parameter-pairs'**. An *action-parameter-pair* is an **action-parameter-id** paired with a corresponding **parameter-value**.

   1) Valid *action-parameter-ids* are specified in the corresponding managed object class.

   2) The representation of this information item is technology-specific. The invoker must supply:

i) *action-parameter-id* according to the parameter naming conventions of the technology employed;

ii) *value* according to type/value representation of the technology employed.

3.4.2.4.3 ACTION Result Primitive Information Items

a) **performer-credentials**. See 3.3.5.1.3.

b) **invoke-id**. See 3.3.5.1.2.

c) **primitive-kind**. (= ‘result’) See 3.3.5.1.2.

d) **completion-status**. This item reports the degree to which the performer was able to perform the action as of the time the result was returned. Possible values for this item are:

1) ‘Acknowledged’. The receipt of the ACTION invocation is acknowledged by the target managed object, but the determination of whether or not the operation can be successfully performed must be deferred.

2) ‘Accepted’. The ACTION invocation has been accepted and the operation is guaranteed to be performed.

3) ‘Completed’. The ACTION operation has been performed successfully. The return of a *completion-status* with value ‘acknowledged’ or ‘accepted’ requires that the performer subsequently invoke either an *action-success* or *action-failure* NOTIFY operation (see 3.4.2.5) when the action has either succeeded or failed, respectively.

NOTE – The need for and the ability to defer the performance of an action may depend on the general capabilities of an SLE Complex and on its implementation of individual actions. In any case, the invoker shall be prepared to process and react to any value of *completion-status*.

3.4.2.4.4 ACTION Error Primitive Information Items

a) **performer-credentials**. See 3.3.5.1.3.

b) **invoke-id**. See 3.3.5.1.2.

c) **primitive-kind** (= ‘error’). See 3.3.5.1.2.

d) **diagnostic**

1) ‘Unable to comply’. See 3.3.5.1.4.

2) ‘invoker unknown’. See 3.4.2.1.4.

3) ‘access denied’. See 3.3.5.4.4.

4) ‘target-mo unknown’. See 3.4.2.1.4.

5) ‘operation not supported’. See 3.4.2.1.4.

6) ‘invocation disabled in current state’. See 3.4.2.3.4.

7) ‘invalid action-id’. An invalid *action-id* was requested for the subject managed object.

8) ‘unknown parameter’. An *action-parameter-id* did not apply to the requested action.

9) ‘missing parameter’. One or more *action-parameter-ids* are missing from the invocation.

10) ‘invalid parameter value’. A *parameter-value* did not match the data type implied by its paired *action-parameter-id.*

11) ‘other’. See 3.3.5.1.4.

NOTE – Further *diagnostics* are described with actions in the respective managed object class.

3.4.2.5 NOTIFY

3.4.2.5.1 General

a) **Purpose.** The NOTIFY operation causes SLE Utilization Management to receive and process a notification issued by the notifying managed object.

b) **Invoker.** This operation can be invoked by the originating managed objects or by SLE Complex Management on behalf of the originating managed object in order to issue a notification.

c) **Performer.** If a telecommunication association with SLE Utilization Management exists, the NOTIFY operation is performed by a notification processing function within the UM-application.

   1) If a telecommunication association with SLE Utilization Management does not exist and a notificationLog managed object exists, the CM-application shall record the notification for deferred retrieval via the notificationLog managed object (see 5.9).

   2) If no notificationLog managed object is available, the notification shall be ignored.

   3) In either case, the invoker shall not become aware of any difference and it is assumed that the CM application supplies an appropriate result to the invoker.

3.4.2.5.2 NOTIFY Invocation Primitive Information Items

a) **invoker-credentials**. See 3.3.5.1.2.

b) **invoke-id**. See 3.3.5.1.2.

c) **operation-type** (= 'NOTIFY'). See 3.4.2.1.2.

d) **primitive kind** (= 'invocation'). See 3.3.5.1.2.

f) **notification-message-type.** This identifies the type of notification that is being issued.

g) **notification-message**. This is the actual notification as it is defined in 4.9.3.

   1) The structure of this message must conform to the *notification-message-type*.

   2) The representation of this item is technology-specific and must be formed according to the type/value system of the technology employed. These rules are specified in reference [9].

3.4.2.5.3 NOTIFY Result Primitive Information Items

a) **performer-credentials**. See 3.3.5.1.3.

b) **invoke-id**. See 3.3.5.1.2.

c) **primitive-kind** (= 'result'). See 3.3.5.1.2.

3.4.2.5.4 NOTIFY Error Primitive Information Items

a) Upon receipt of a NOTIFY error primitive, the CM-application shall log the original notification message for deferred access via the notificationLog managed object (see 5.5).

b) **performer-credentials**. See 3.3.5.1.3.

c) **invoke-id**. See 3.3.5.1.2.

d) **primitive-kind** (= 'error'). See 3.3.5.1.2.

e) **diagnostic**

   1) 'unable to comply'. See 3.3.5.1.4.

   2) 'originating mo unknown'. The performer cannot identify the originating managed object.

   3) 'access denied'. See 3.3.5.4.4.

   4) 'operation not supported'. See 3.4.2.1.4.

5) ‘notification message type error’. The notification type does not exist or does not belong to the originating managed object’s class.

6) ‘notification message garbled’. The *notification-message* does not conform to the *notification-message-type*.

7) ‘other’. See 3.3.5.1.4.

3.4.2.6 CREATE

3.4.2.6.1 General

a) **Purpose**. The CREATE operation causes the construction of a new managed object as an instance of a specified managed object class.

b) **Invoker**. A UM-application can invoke this operation.

c) **Performer**. A CM-application performs the operation.

NOTE – This draft Recommendation does not prescribe how the creation capability is implemented. For CREATE, the CM-application needs an instantiation capability. This capability may be allocated to the target managed object or to a dedicated ‘object factory’.

3.4.2.6.2 CREATE Invocation Primitive Information Items

a) **invoker-credentials**. See 3.3.5.1.2.

b) **invoke-id**. See 3.3.5.1.2.

c) **operation-type** (= ‘CREATE’). See 3.4.2.1.2.

d) **primitive-kind** (= ‘invocation’). See 3.3.5.1.2.

e) **target-mo-name**. See 3.4.2.1.2. In this case, the *target-mo-name* is the managed object instance that shall contain the new managed object.

f) **mo-class**. This information item identifies the managed object class from which the new managed object shall be instantiated. The representation of this information item is technology-specific. The invoker must supply it according to the translation rules for the translation of Object Identifiers (OID) to the type reference mechanism of the technology employed.

g) **ref-mo**. The *ref-mo* identifies a reference managed object from which the attribute values shall be copied into the new managed object.

   1) The representation of this information item is technology-specific. The invoker must supply it according to the translation rules for the translation of DNs to the instance reference mechanism of the technology employed.

2) Only the *ref-mo* itself shall be copied; any objects contained by the *ref-mo* shall not be copied.

3) The reference managed object must be of the managed object class designated by *mo-class*.

4) The copy of the reference managed object shall have all attributes that can be set by the UM-application during the definition phase set to their current values at the time when the CREATE operation is performed. Values of read-only attributes shall be left unspecified.

5) This item is optional.

NOTES

1 Creation with a reference managed object indicates that a new managed object is 'the same as' the reference managed object—for instance, that a notificationLog managed object in servicePackage B is identical to one already specified in servicePackage A.

2 At any time between its creation and deletion, any managed object in the realm of the UM-application may be selected as a *ref-mo*.

i) A set of **'attribute-value-pairs'**. An *attribute-value-pair* shall consist of an **attribute-id** paired with its *value* (*attribute-id, value*), where *attribute-id* is the name of an attribute defined in the managed object's class, and *value* is a value for this attribute taken from the data type and range defined for the specified attribute.

1) This item is mandatory. It must contain at least one (*attribute-id, value*) pair for the naming attribute.

2) If this item is present and a *ref-mo* is specified, the values specified shall override those taken from the *ref-mo*.

3) The set of *attribute-value-pairs* in a CREATE invocation <u>may</u> cover attributes that can be set by SLE Utilization Management during the definition phase. If one or more of these attributes are not specified at creation time, they must be set before validation time; otherwise a 'completeness defect' shall be added to the set of *defect-records* of the associated servicePackage managed object (see 5.17.5).

4) The set of *attribute-value-pairs* in a CREATE invocation <u>must not</u> cover attributes that are read-only for SLE Utilization Management.

5) The representation of this item is technology-specific. The invoker must supply:

   i) *attribute-id* according to the attribute naming conventions of the technology employed.

   ii) *value* according to type/value representation in of the technology employed.

3.4.2.6.3 CREATE Result Primitive Information Items

a) **performer-credentials**. See 3.3.5.1.3.

b) **invoke-id**. See 3.3.5.1.2.

c) **primitive-kind** (= 'result'). See 3.3.5.1.2.

3.4.2.6.4 CREATE Error Primitive Information Items

a) **performer-credentials**. See 3.3.5.1.3.

b) **invoke-id**. See 3.3.5.1.2.

c) **primitive kind** (= 'error'). See 3.3.5.1.2.

d) **diagnostic**

   1) 'unable to comply'. See 3.3.5.1.4.

   2) 'invoker unknown'. See 3.4.2.1.4.

   3) 'access denied'. See 3.3.5.4.4.

   4) 'target-mo unknown'. See 3.4.2.1.4.

   5) 'operation not supported'. See 3.4.2.1.4.

   6) 'invalid MO class'. An invalid managed object class was specified for *mo-class*.

   7) 'unknown containing object'. The object identified by *containing-mo* did not exist.

   8) 'containment violation'. The requested *mo-class* cannot be contained by the *containing-mo* according to the rules identified in SLE managed object containment.

   9) 'precondition violation'. Creation of a managed object may be constrained by a number of preconditions. Such preconditions are defined in the subsections on creation aspects for each SLE managed object.

   NOTE – This draft Recommendation assumes that the performer is able to verify these preconditions and reject a CREATE invocation if one or more of them are not met.

   10) 'unknown ref-mo'. A non-existent reference managed object was specified.

   11) 'ref-mo mismatch'. *Mo-class* and the class of *ref-mo* do not match.

   12) 'unknown attribute'. An *attribute-id* was not recognized by the subject managed object. The *diagnostic* contains the additional information: *attribute-id* of the unknown attribute.

13) 'invalid attribute value'. An *attribute-value* did not match attribute data type.

14) 'no name for new MO'. A value for the naming attribute of the MO to be created is not specified.

15) 'duplicate name for new MO'. The name for the MO to be created has already been used.

16) 'out of scope of service agreement'. The requested managed object was of a class not supported by the Service Agreement under which the object was requested.

17) 'other'. See 3.3.5.1.4.

NOTES

1 The bounds imposed by Service Agreements and the ability of SLE Complex Management to automatically detect violations of those bounds are matters local to specific SLE Complex Management and SLE Utilization Management implementations.

2 An example of this *diagnostic* being returned is as the result of an attempt to create an r-af-ts-p managed object when only return channel frames services were negotiated for the applicable capability set.

3 Other violations of Service Agreement constraints may not be detectable at managed object creation time, but only detected in service package validation. For example, an attempt to create a tenth instance of an r-cf-ts-p managed object when the Service Agreement limits any single service package to nine instances, may result in a 'resource limitation defect'.

3.4.2.7 DELETE

3.4.2.7.1 General

a) **Purpose**. The DELETE operation causes the deletion of a specified managed object instance and the release of all associated Complex resources.

b) **Invoker**. A UM-application can invoke this operation.

c) **Performer**. A CM-application can perform this operation.

3.4.2.7.2 DELETE Invocation Primitive Information Items

a) **invoker-credentials**. See 3.3.5.1.2.

b) **invoke-id**. See 3.3.5.1.2.

c) **operation-type** (= 'DELETE'). See 3.4.2.1.2.

d) **primitive-kind** (= 'invocation'). See 3.3.5.1.2.

e) **target-mo-name**. This item identifies the managed object to be deleted. For further details see 3.4.2.1.2.

3.4.2.7.3 DELETE Result Primitive Information Items

a) **performer-credentials**. See 3.3.5.1.3.

b) **invoke-id**. See 3.3.5.1.2.

c) **primitive-kind** (= ‘result’). See 3.3.5.1.2.

3.4.2.7.4 DELETE Error Primitive Information Items

a) **performer-credentials**. See 3.3.5.1.3.

b) **invoke-id**. See 3.3.5.1.2.

c) **primitive-kind** (= ‘error’). See 3.3.5.1.2.

d) **diagnostic**

   1) ‘unable to comply’. See 3.3.5.1.4.

   2) ‘invoker unknown’. See 3.4.2.1.4.

   3) ‘access denied’. See 3.3.5.4.4.

   4) ‘target-mo unknown’. See 3.4.2.1.4.

   5) ‘operation not supported’. See 3.4.2.1.4.

   6) ‘other’. See 3.3.5.1.4.

3.4.2.8 DISCOVER

3.4.2.8.1 General

a) The implementation of this operation shall be optional.

b) **Purpose**. The DISCOVER operation returns a set of the names of all managed objects immediately contained by a selected target managed object. By iterative use of DISCOVER, the complete containment tree can be explored.

NOTE – This operation is intended to help SLE Utilization Management reconstruct the containment tree, for instance in response to a UM-application crash.

c) **Invoker**. A UM-application can invoke this operation.

d) **Performer**. A CM-application can perform this operation.

3.4.2.8.2 DISCOVER Invocation Primitive Information Items

a) **invoker-credentials**. See 3.3.5.1.2.

b) **invoke-id**. See 3.3.5.1.2.

c) **operation-type** (= 'DISCOVER'). See 3.4.2.1.2.

d) **primitive-kind** (= 'invocation'). See 3.3.5.1.2.

e) **target-mo-name**. This item identifies the managed object for which the contained managed objects shall be identified. For further details see 3.4.2.1.2. The representation of this item is technology-specific. The identification of the managed objects must be constructed according to the applicable technology mapping rules.

3.4.2.8.3 DISCOVER Result Primitive Information Items

a) **performer-credentials**. See 3.3.5.1.3.

b) **invoke-id**. See 3.3.5.1.2.

c) **primitive-kind** (= 'result'). See 3.3.5.1.2.

d) A set of '**mo-ids'**. Each element in the set shall identify a managed object that is immediately contained by the managed object identified by the *target-mo-name*. The set is exhaustive and shall cover all managed objects immediately contained. If the managed object identified by *target-mo-name* does not contain any managed objects, the set shall be empty. The representation of this item is technology-specific. The identification of the managed objects must be constructed according to the applicable technology mapping rules.

3.4.2.8.4 DISCOVER Error Primitive Information Items

a) **performer-credentials**. See 3.3.5.1.3.

b) **invoke-id**. See 3.3.5.1.2.

c) **primitive-kind**. (= 'error') See 3.3.5.1.2.

d) **diagnostic**

   1) 'unable to comply'. See 3.3.5.1.4.

   2) 'invoker unknown'. See 3.4.2.1.4.

   3) 'access denied'. See 3.3.5.4.4.

   4) 'target-mo unknown'. See 3.4.2.1.4.

   5) 'operation not supported'. See 3.4.2.1.4.

   NOTE – The implementation of the behavior for this operation is optional.

   6) 'other'. See 3.3.5.1.4.

3.4.3 FUNCTIONAL REQUIREMENTS TO SUPPORT OPERATIONAL INTERACTION

To support the management operations described in 3.4.2, a channel must have the following capabilities:

a) With regard to networking, a channel shall be able to provide network connectivity between a node hosting a CM-application and a remote node hosting a UM-application.

b) With regard to telecommunications, a channel shall be able to:

   1) establish, suspend, resume, and release a telecommunication association between the nodes;

   2) signal telecommunication events to managed object implementations;

   3) encode and decode protocol data units according to a transmission syntax and generate related error primitives;

   4) construct service primitives from protocol data units, and vice versa.

c) With regard to support of managed objects, a local channel infrastructure shall be able to:

   1) Verify a service primitive with respect to the applicable type/value system for completeness and compatibility, and generate the related error returns.

   2) Resolve names (i.e., identify the local implementation of a target managed object on the basis of a name found in a service primitive) and generate related error primitives.

   3) Construct names (i.e., generate a technology-specific representation on the basis of the containment relationship between managed objects).

   4) Support CREATE and DELETE operations for managed objects. This includes a capability to specify a name for a new managed object and to manage (establish, maintain and release) the containment relationship between the MOs in a target group.

   5) Support all operational interaction between the UM-application and the CM-application: GET, SET, ACTION, NOTIFY and DISCOVER.

   6) Support security mechanisms according to the agreed security measures. This includes:

      i) verification of credentials based on information stored in the Security Information Base;

      ii) restriction of access to a single target-group, which is identified by a serviceAgreement managed object;

iii) generation of related error returns.

d) With regard to direct interaction with the managed object implementation, a channel shall be able to:

1) Activate and deactivate a managed object and generate related error returns (e.g., with *diagnostic* 'unable to comply');

2) marshal data structures between the programming environment of the channel and the programming environment of the managed object implementation, and generate related error returns;

3) support persistence of managed object implementations.

NOTE – Typically, state-of-the-art COTS middleware provides a large range of functions and services, which presumably cover most of the capabilities identified in (1) through (3) above. Nevertheless, it may be possible that a capability is not provided off-the-shelf. In this case, a trade-off shall decide whether:

– the demanded capability will be implemented as an add-on (to the technology or to the application);

– the capability requirement will be relaxed;

– the capability requirement will be dropped.

The mapping rules for the respective technology must reflect the trade-off made and the decision.

4 GENERAL CHARACTERISTICS OF SLE MANAGED OBJECTS

4.1 GENERAL

This section specifies features common to all SLE service management managed objects.

4.2 INHERITANCE

The SLEManagementEntity managed object class is the parent class from which all other SLE managed object classes are derived. All SLE managed objects inherit the properties of the SLEManagementEntity managed object class, as follows:

a) Figure 4-1 shows the managed object classes that are derived from SLEManagementEntity.

b) Figure 4-2 shows the managed object classes that are derived from the ChannelStored managed object class.

c) Figure 4-3 shows the managed object classes that are derived from the Production managed object class.

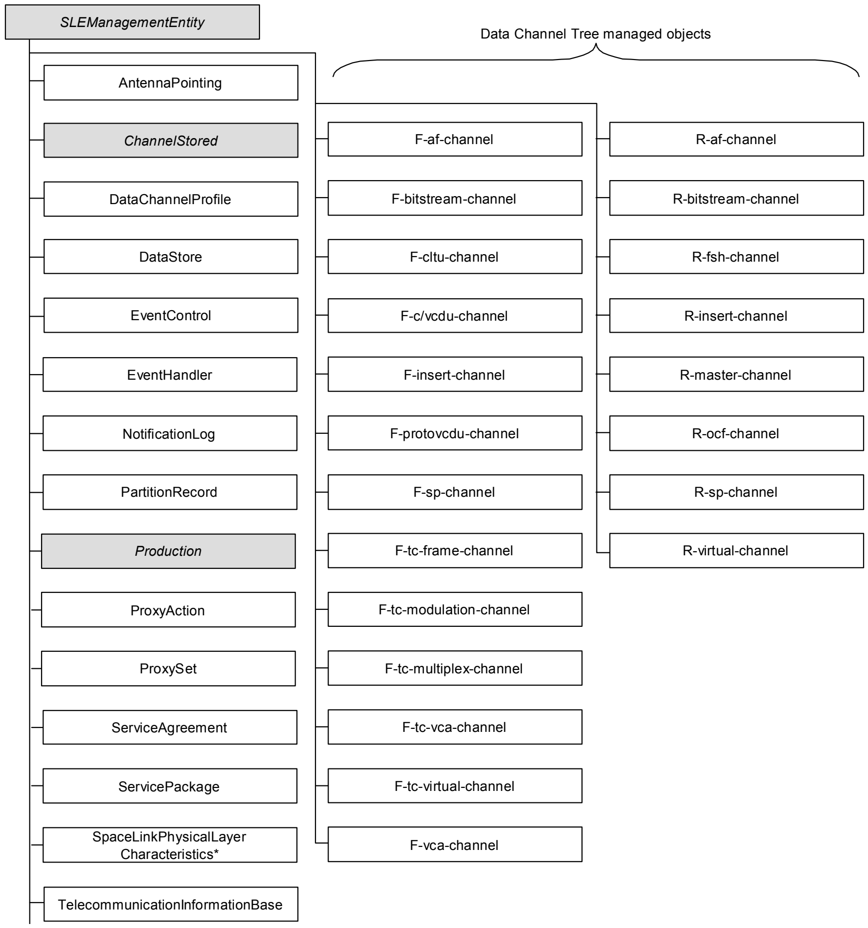
d) Figure 4-4 shows the managed object classes that are derived from the TransferService managed object class.

e) Classes derived from the Authentication managed object class are depicted and defined in reference [12].

NOTES

1 Abstract classes appear as shaded boxes; concrete classes appear as white boxes in all inheritance diagrams in this draft Recommendation.

2 These diagrams only depict classes defined in this draft Recommendation. Other Recommendations in the Service Management suite will depict the inheritance of classes defined therein.



*Specified in reference [13].

Figure 4-1: SLEManagementEntity Derived Managed Objects

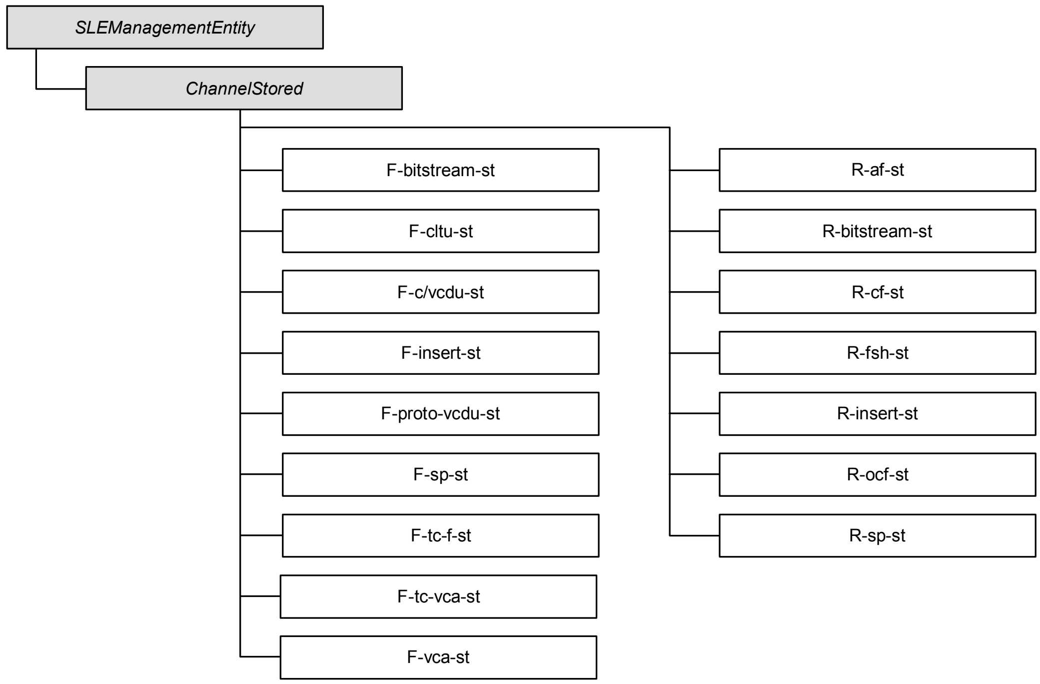


Figure 4-2: ChannelStored Derived Managed Objects

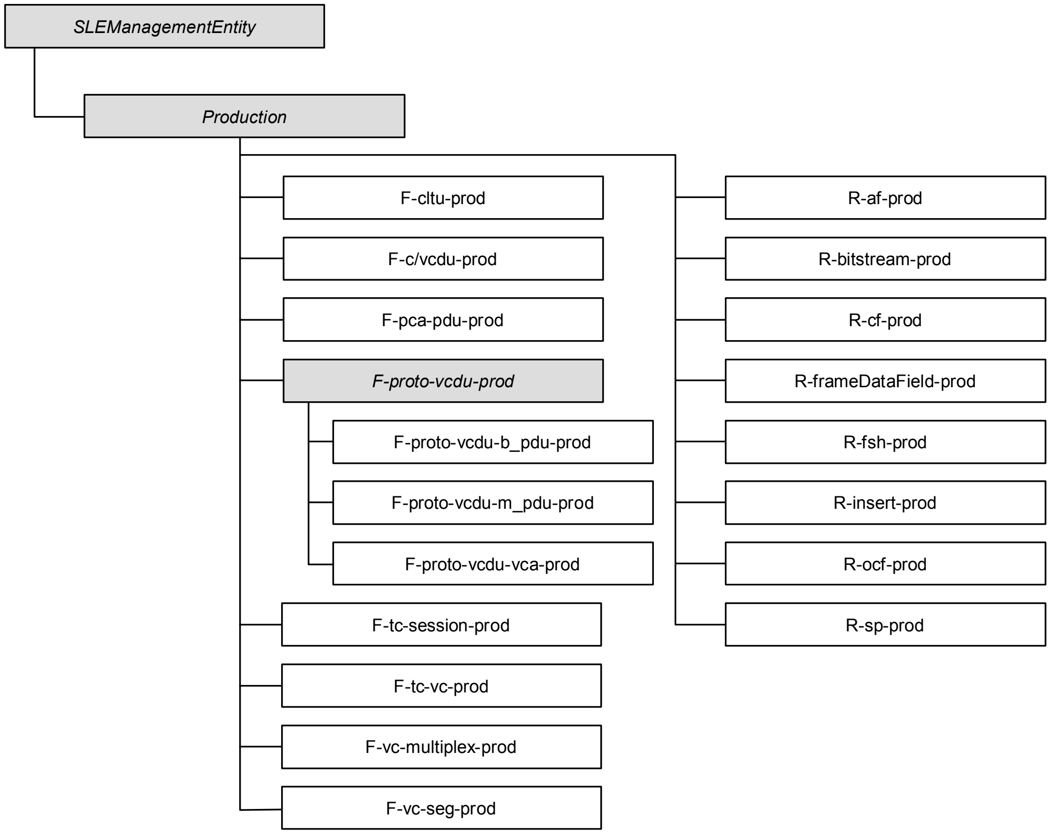


Figure 4-3: Production Derived Managed Objects

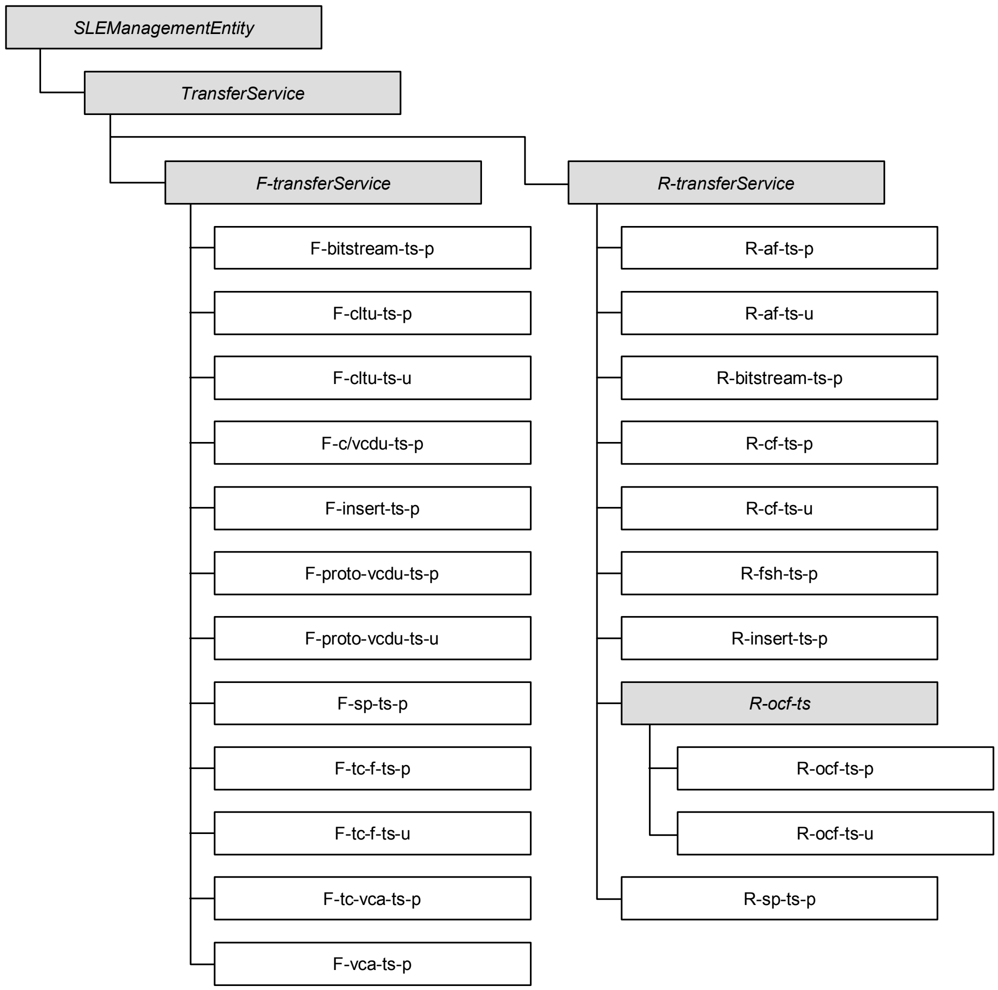


Figure 4-4: TransferService Derived Managed Objects

4.3 OBJECTS CONTAINED

a) A given managed object instance may contain one or more instances of other managed object classes.

b) The preconditions and constraints for containment of a managed object instance are specified in the description of the containing managed object class.

4.4 CREATION

a) SLE Utilization Management shall create a managed object instance by invoking a CREATE operation, which identifies the desired managed object class and specifies its containing managed object. The CREATE operation is specified in 3.4.2.6. SLE Complex Management checks the correctness of the invocation, and ensures that the applicable constraints and dependencies are fulfilled.

b) Creation shall take place only in a container object.

c) At creation time, a value must be specified for each managed object attribute. Attribute values may be specified in parameters of the CREATE operation, or may be copied from an existing reference managed object instance of the same class.

d) After creation, a new managed object is in state ‘waiting’.

e) A successful creation ensures that the new managed object conforms to the present draft Recommendation.

NOTES

1 Successful creation does not ensure that the managed object is internally consistent or that it is not in conflict with other managed objects. Only when the service package is validated is the Managed object checked for internal consistency and for potential conflicts.

2 Successful creation does not imply that the Complex’s resource, which the managed object represents, is already allocated and associated with the managed object. Resource allocation and association may happen later in the managed object’s lifetime. This event is reflected in an appropriate state transition (e.g., from ‘validated’ to ‘committed’).

4.5 DELETION

a) SLE Utilization Management shall delete a managed object instance by invoking a DELETE operation, which identifies the managed object instance to be deleted. The DELETE operation is specified in 3.4.2.7.

b) If a managed object is deleted, all its contained managed objects shall be deleted as well. In this case, possible ongoing resource activities, which are represented by such contained MOs, shall be aborted.

c) A managed object instance can be deleted if it is in the state ‘waiting’, or in the state ‘done’. Specific preconditions and constraints for deletion are specified in the managed object’s detailed description of states and state transitions.

4.6 MONITORING AND COMMANDING

a) A managed object shall offer SLE Utilization Management the following means for monitoring:

   1) inspection of the current value of managed object attributes;

   2) inspection of the managed object’s current containment tree;

   3) issuance of a notification.

b) A managed object shall offer SLE Utilization Management the following means for commanding:

1) setting the managed object's attribute values;

2) invoking an action.

c) The related operations (GET, SET, ACTION, NOTIFY, DISCOVER) are defined in 3.4 of this draft Recommendation.

4.7 ATTRIBUTES

4.7.1 GENERAL

a) An attribute reflects a property of the associated resource that is of interest to SLE Utilization Management.

b) An attribute may be a conditional or mandatory element of a managed object. A conditional attribute and the applicable conditions are explicitly identified in its respective description. All other attributes shall be considered mandatory elements of the managed object.

c) After creation of a managed object instance, all its attributes shall assume an initial value.

d) SLE Utilization Management may inspect any attribute's value using a GET operation, which is defined in 3.4.2.2.

e) SLE Utilization Management may modify an attribute's value using a SET operation, which is defined in 3.4.2.3.

   1) In the definition phase, when the managed object is in state *waiting*, all its attributes may be SET.

   2) During the utilization phase, setting of attribute values is allowed for only a few attributes. Attributes that shall be settable during the utilization phase are marked as such in their respective descriptions.

   3) The confirmation of a SET operation with a SET-result indicates to Utilization Management that the associated resource <u>is</u> reconfigured accordingly. The confirmation of a SET operation with a SET-error indicates to Utilization Management that the associated resource is not affected.

   4) Setting of an attribute value may be enabled or disabled according to the actual value of some attribute in another managed object. Such a dependency is part of the attribute's behavior description.

NOTE – A typical case is that an attribute can be modified only if a containing managed object is in a certain state. For example, the *event-handler-state* attribute of an eventHandler managed object may be set to *enabled* only if the servicePackage managed object that contains it is *executing*.

5) Setting of an attribute value may be constrained to a range of values, which is specified in another Managed object. Such a constraint is part of the attribute's behavior description.

f) The right to inspect an attribute's value does not include the right to set its value. However, the right to set an attribute value shall include the right to inspect its value.

NOTES

1 The attribute subsection for each managed object defines that managed object's specific (as opposed to inherited) attributes. These attributes and any inherited attributes are identified in an attribute value summary table that reviews the complete list of that managed object's attributes. The table indicates whether SLE CM or UM supplies the initial attribute value, when that value is specified, who may modify it, and its initial value. In these tables, '.inv' indicates that the initial value is specified as part of an operation invocation; '.rtn' indicates that the value is returned by the operation. A dash (—) indicates that the value is agreed upon by a means outside the scope of this draft Recommendation. If there are time constraints on when attribute values may be changed, those constraints are identified in the modifications column.

2 In the attribute definitions in the remainder of this draft Recommendation, any dependencies on other attributes or on states are identified in the attribute definition. If no dependencies are identified, it is assumed that there are none.

4.7.2 OPTIONAL ATTRIBUTES

a) Some managed object attributes are qualified as 'optional'. Such an attribute may or may not be supported by the Complex Management.

b) If an optional attribute is 'supported' by SLE Complex Management, it shall behave exactly like a mandatory attribute with respect to behavior and SET and GET operations.

c) If an optional attribute is 'not supported' by SLE Complex Management, the following shall apply:

1) the Complex shall not implement the respective behavior;

2) a GET invocation shall trigger a GET-error with *diagnostic* 'operation not supported';

3) a SET invocation shall trigger a SET-error with *diagnostic* 'operation not supported';

4) a CREATE operation shall succeed irrespective of the presence or absence of values for an optional attribute.

NOTE – At the expense of a relatively low overhead for the rudimentary support of a ‘not supported’ optional attribute, this approach allows:

- SLE Utilization management to use the same application software to cooperate with different SLE Complex Managements, even if these support different sets of optional attributes;
- SLE Complex Management to (almost) ignore attributes for which support would be difficult or costly to implement (i.e., a Complex that does not implement an optional attribute must nevertheless implement the behavior described above);
- SLE Management to extend the set of supported optional attributes without imposing software adaptations to customer Utilization Managements.

4.8 ACTIONS

a) SLE Utilization Management uses an ACTION operation to effect a change of behavior of the resource that the target managed object represents. This kind of change is identified by the name of the action invoked; the details of the change are specified by a set of parameter values, which are supplied in the ACTION invocation. The ACTION operation is described in detail in 3.4.2.4.

b) For a given Managed object and a given state, an ACTION invocation shall either succeed, fail, or be ignored. ACTION invocations that succeed or fail are identified in the state-related behavior portion of each managed object’s description. All other ACTION invocations shall be ignored. An ACTION operation may succeed or fail depending on the state of the managed object. This behavior is described in the subsection on the state-related behavior.

c) The managed object must confirm each ACTION invocation with either an ACTION-result or ACTION-error returned to the invoker. The precise meaning of the information returned is described as part of the action definition.

NOTE – The meaning of an ACTION-result may range from promising to perform the action to indicating the completion of the action on the resource level.

4.9 NOTIFICATIONS

4.9.1 GENERAL

a) The NOTIFY operation is described in 3.4.2.5.

b) A managed object may issue a notification at any time.

c) If a telecommunication association exists, all notifications shall be sent directly to SLE Utilization Management. If a telecommunication association does not exist (SLE Complex Management and SLE Utilization Management are unbound), notifications shall be logged for deferred retrieval via a notificationLog managed object. Subsection 5.9 defines the NotificationLog managed object class, which allows users to access recorded notifications.

d) A managed object's state transition may trigger a state transition notification. The state transitions that cause a notification to be issued are identified in the subsection on state-related behavior.

NOTE – In order to avoid generating an overload of notifications, not all state transitions result in notifications.

4.9.2 BEHAVIOR OF NOTIFICATIONS

To avoid an overload of notifications, a managed object shall apply the following suppression policy:

a) A managed object that may issue notifications associates a *notification-timeout*, a *suppressed-notifications-counter* and a *flag* (notification issuing enabled/disabled) with each type of its notifications. For all types of notifications, the *notification-timeout* has the same duration. It is a Complex-specific constant, which is fixed in the attribute *notification-timeout* of the applicable serviceAgreement managed object.

b) If a managed object enters the state *configured,* all its *suppressed-notification-counters* shall be set to zero and issuing notifications shall be enabled.

c) If the complex-internal precondition for a notification is fulfilled and issuing is enabled, the managed object shall issue a notification.

d) The notification shall carry the current value of the respective *suppressed-notification-counter*.

e) After a notification has been issued, the respective *suppressed-notification-counter* is set to zero.

f) After a notification has been issued, the *notification-timeout* period shall start.

g) During the notification timeout period, issuing of notification of the respective type shall be disabled.

h) If, during the timeout period, the precondition for a notification is fulfilled again, the type-specific *suppressed-notification-counter* shall be incremented by 1, but a notification shall not be issued.

i) Immediately after the respective *notification-timeout* has expired, issuing of notifications of the respective type shall be enabled again.

NOTE – In subsection 4.9.2, the term ‘generates a notification’ is used in two different situations:

- First, when a telecommunication association with SLE Utilization Management exists, a NOTIFY invocation is issued. In this case, the relevant information appears in the NOTIFY invocation as *notification-message.*
- Second, if a telecommunication association with SLE Utilization Management does not exist, SLE Complex Management logs the notification information for deferred retrieval via a notificationLog managed object. In this case, the return to a *dump* action carries the relevant *notification-message* in a *notification-record*.

4.9.3 CONTENTS OF NOTIFICATION MESSAGES

4.9.3.1 General

a) A *notification-message* is a constituent of the NOTIFY invocation (see 3.4.2.5.2 and a constituent of *notification-record* as dumped via a notificationLog managed object (see 5.9).

b) A *notification-message* is comprised of the following items:

   1) **originator-mo**. The *originator-mo* identifies the managed object that issues the notification. The representation of this item is technology-specific and must be provided according to the mapping rules defined for Distinguished Names in to the applicable implementation technology (reference [9]).

   2) **issue-time**. The originator managed object encountered the precondition for issuing the notification at the *issue-time*.

      i) Data type/value: CCSDS time code.

   3) **suppressed-notification-counter**. The *suppressed-notification-counter* is the number of suppressed notifications since the last notification of the same type was issued (see 4.9.2).

      i) Data type/value: Unsigned integer.

   4) **specific notification items**. Subsection 4.9.3.2 defines specific notification items for state transition notifications. Other specific notification items may be found in the originator managed object class definitions.

4.9.3.2 Specific Information Items of State Transition Notifications

A state transition notification shall include the following specific notification items:

a) **from-state**. The managed object existed in the *from-state* when the precondition for the state transition was fulfilled. Data type/value: The value of the state attribute for the *originator-mo*.

b) **to-state**. After the state transition, the managed object exists in the *to-state*. Data type/value: The value of the state attribute for the *originator-mo*.

4.10 STATES

NOTE – Many managed objects feature a state attribute. During the definition phase, this state attribute reflects the progress of the planning activity. During the utilization phase, the state attribute reflects availability and operational conditions of the associated Complex-internal resource.

a) A state transition may be caused by the performance of an action, by a Complex-internal event, or by the passage of time.

b) A managed object shall enter the state *done* if all its contained managed objects have entered the state *done*.

c) A managed object shall be considered active while it is associated with its corresponding resource.

Table 4-1: Active States

Managed Object Class	Active State(s)
ChannelStored	*configured*
DataStore	*configured*
EventHandler	*configured*, *enabled* and *disabled*
Production	*configured* and *operational*
AntennaPointing	*configured* and *active*
SpaceLinkCarrierPackage	*configured* and *operational*
TransferService	*configured*, *ready* and *active*

4.11 ASSOCIATION WITH COMPLEX RESOURCES

a) SLE Complex Management shall be responsible for associating a managed object instance with one or more Complex-internal resources for space link acquisition and production or delivery of SLE transfer services.

b) This draft Recommendation relies on the following properties of this association during the definition phase: A managed object shall be associated with SLE Complex Management's planning resources.

c) This draft Recommendation relies on the following properties of this association during the utilization phase:

1) A managed object shall be associated with a SLE Complex's functional resources. This association happens when the managed object enters the state *committed*.

2) A current attribute value of the managed object is derived from a corresponding measurement at one or more Complex-internal resources. The attribute value is updated such that it always reflects—within a Complex-specific time window—the actual value at the corresponding measurement point.

3) The value to which an attribute is SET is carried forward—within a Complex-specific time window—to a suitable resource within the Complex.

4) A successful ACTION operation takes effect—within a Complex specific time window—on the appropriate Complex-internal resources.

5) A GET operation provides—within a Complex-specific time window—the latest value of the associated resource.

6) An information item in a notification message reflects—within a Complex-specific time window—the latest values of the associated resource.

7) Complex-internal preconditions for issuing notifications match with the intent of the notifications as described in the managed objects class definitions.

NOTES

1 This draft Recommendation neither prescribes nor constrains the implementation of these associations by SLE Complex Management.

2 This draft Recommendation neither prescribes nor constrains the length of the various Complex-specific time windows mentioned above.

3 The length of the Complex-specific time windows may be an important quality-of-service characteristic for the selection of an SLE Complex by a customer.

4.12 PERSISTENCE

For SLE Utilization Management, a managed object appears persistent throughout its lifetime, from creation until deletion.

4.13 SECURITY

a) A managed object at the SLE Service Management Interface shall be accessible exclusively to the SLE Utilization Management that created it.

b) Exclusive access shall be ensured by the applicable authentication and access control measures that are agreed between SLE Utilization Management and SLE Complex Management, and laid down as part of the binding information.

c) Access to a managed object shall be denied if the demands of the applicable security measures are not met.

d) A managed object shall be accessible to SLE Complex Management by means that are outside the scope of this draft Recommendation.

4.14 SUPPORT OF DEBRIEFING

With respect to providing SLE transfer services, mission organizations' debriefing requirements may vary considerably and are difficult to standardize. Therefore this draft Recommendation adopts the following approach:

a) Managed objects include a number of attributes that are of potential interest for debriefing reports.

b) These attributes are 'read-only and 'optional' (see 4.7).

c) At any time, SLE Utilization Management can retrieve the desired information and use it for a custom-built debriefing report via a series of GET operations to selected managed objects and to the desired attributes therein.

d) SLE Complex Management guarantees that when the managed object enters the state *done*, the attribute shall be supplied with the actual value acquired from the associated resource.

4.15 VALIDATION

a) If a given managed object does not pass validation, neither the managed object nor its contained managed objects can be used for Complex resource allocation.

b) For each servicePackage managed object, SLE Complex Management shall perform a validation process, which ensures:

   1) completeness of the specification;

   2) internal consistency of the specification;

   3) compatibility of the managed object instance with other related managed object instances;

   4) absence of resource usage conflicts with other servicePackages.

c) This draft Recommendation assumes that the duration of a validation process is a Complex-specific characteristic and does not exceed a predictable maximum.

NOTES

1 This draft Recommendation neither prescribes nor constrains the way in which SLE Complex Management realizes these assumptions.

2 The depth of the validation may be an important quality-of-service characteristic for the selection of a Complex by a customer.

3 The duration of a validation process may be an important quality-of-service characteristic for the selection of a Complex by a customer.

4.16 REGISTRATION OF MANAGED OBJECT CLASSES

a) For the formalization of this narrative description in the (reference [10]), it must be possible to unambiguously reference each type of information item identified above. This is achieved by 'registering' the relevant items by assigning so-called Object Identifiers (OID).

b) The concept of registration is discussed in annex C.

4.17 NAMING OF MANAGED OBJECT INSTANCES

a) Each SLE Managed object instance is named by a unique Distinguished Name (DN), which shall be constructed according to the rules defined in annex C.

NOTE – For the construction of its DN, each managed object class xxx has a dedicated naming attribute (*xxx-mo-id*).

b) The containment relationships specified in the individual managed object class descriptions shall form the basis for naming managed object instances.

5 COMMON AND ABSTRACT MANAGED OBJECT DEFINITIONS

5.1 AntennaPointing MANAGED OBJECT CLASS

5.1.1 PURPOSE

5.1.1.1 General

An antennaPointing managed object shall:

a) identify the operationally applicable trajectory prediction from the set of predictions available to a mission;

b) enable a Complex resource to track the spacecraft according to the applicable trajectory prediction. SLE Utilization Management shall download trajectory predictions as data files.

5.1.1.2 Perception by SLE Utilization Management

a) Trajectory prediction information must be provided if the Complex eventually performs space link acquisition.

b) This information shall be generated by a trajectory and attitude computation entity within the MDOS.

c) The MDOS may have prepared several trajectory prediction files (for example, the nominal prediction together with alternate predictions covering possible trajectories after certain maneuvers). However, at a given time, one and only one trajectory prediction shall apply.

d) SLE Utilization Management shall be responsible for:

   1) providing trajectory prediction information files according to a format that has been agreed in the *trajectory-format* attribute of the *applicable* serviceAgreement managed object;

   2) coordinating creation of the antennaPointing managed object with the transfer of a suitable trajectory prediction information file;

   3) submitting a trajectory prediction information file in time and with the accuracy required to steer the tracking antenna system.

e) Tracking with an obsolete trajectory prediction can be stopped and immediately resumed with new trajectory predictions.

5.1.1.3 Perception by SLE Complex Management

a) The trajectory prediction file for the mission spacecraft can be accessed via the antennaPointing managed object.

b) SLE Complex Management shall be responsible for

   1) deriving the need for the trajectory prediction file from analysis of the service package;

   2) converting the trajectory information delivered into a format that can be used by the Complex's tracking resources;

   3) promptly downloading a trajectory prediction file to the tracking resource (the resource that acquires the space link);

   4) safeguarding the operationally-used trajectory prediction information from any manipulation that may be applied to it.

5.1.1.4 Creation Aspects

Any attempt to create more than one antennaPointing managed object contained by a given servicePackage managed object shall fail with *diagnostic* 'multiple managed objects not allowed'.

5.1.1.5 Deletion Aspects

a) The antennaPointing managed object shall be deleted if the servicePackage managed object that contains it is deleted. At this time, it is assumed that no tracking activities are performed.

b) Subsection 5.1.7 contains information about the states in which an antennaPointing managed object can be deleted.

5.1.2 INHERITANCE—AntennaPointing MO CLASS

This managed object class is derived from, and inherits the properties of, the SLEManagementEntity managed object class.

5.1.3 OBJECTS CONTAINED—AntennaPointing MO CLASS

An instance of this concrete managed object class contains no other managed objects.

5.1.4 STATES—AntennaPointing MO CLASS

a) In the definition phase, an antennaPointing managed object shall exist in one of the following states:

   1) **Waiting.** The antennaPointing managed object has been created, but is not yet associated with a valid trajectory prediction file.

   2) **Under Validation.** The containing servicePackage managed object is undergoing validation. The antennaPointing managed object is static until validation is complete.

3) **Validated.** The antennaPointing managed object has been associated with a valid trajectory prediction file (made available via the attribute *trajectory-file-id*), but the file is not yet ready for use.

b) In the utilization phase, a antennaPointing managed object shall exist in one of the following states:

1) **Ready.** The trajectory prediction file (made available via the attribute *trajectory-file-id*) has been converted into a format that the Complex's tracking resources can use. The file is ready for use.

2) **Active.** The spacecraft trajectory prediction information file made available via the attribute *trajectory-file-id* has been downloaded and is actively used by the Complex's tracking resources.

3) **Done.** The containing servicePackage managed object has transitioned to *done*.

5.1.5 ATTRIBUTES—AntennaPointing MO CLASS

This managed object class shall have the following specific attributes:

a) **antenna-pointing-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (see annex C).

1) Data type/values: Character string.

2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **trajectory-file-id.** The operationally applicable trajectory prediction file in the trajectory filestore. The file specified is used for the Complex's subsequent tracking activities.

1) Data type/values: Character string.

2) Modifications: SET shall be enabled in both the definition and utilization phases.

3) Dependencies:

i) If the file identified by this attribute does not exist, is not a trajectory prediction file, or does not conform to the agreed Recommendation (*recommendation-version-trajectory* in the serviceAgreement managed object), the SET invocation shall be rejected with *diagnostic* 'invalid trajectory prediction file'.

ii) If the set invocation is rejected, the old attribute value shall be retained. If the SET invocation is accepted and the *tracking-state* is *active*, the new setting shall become immediately effective only if a *new trajectory* action is performed. Otherwise, the new trajectory prediction shall become effective only after the ongoing tracking activity has stopped.

c) **time-offset.** The time correction (in seconds) applied to the trajectory elements in the information file when the file becomes active. If the file is already active, the *time-offset* shall not be applied.

   1) Data type/values: Positive or negative integer in the range 0 – 3600. Initial value is ‘0’.

   2) Modifications: SET shall be enabled in both the definition and utilization phases.

d) **antenna-pointing-state**. The current state of the tracking resources within the SLE Complex. The *antenna-pointing-state* is set by SLE Complex Management and can only be read by SLE Utilization Management.

   1) Data type/values: See 5.1.4.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

e) **antenna-pointing-time.** The accumulated time the managed object was in the state *active*.

   1) Data type/values: Unsigned integer. Initial value shall be ‘0’.

   2) Modifications:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

f) **done-time.** The time when the managed object entered the state *done.*

   1) Data type/values: CCSDS time code. Initial value shall be ‘null’.

   2) Modifications:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

Refer to table 5-1 for the antennaPointing managed object attribute value summary.

Table 5-1: Attribute Value Summary—AntennaPointing Managed Object

Attribute	Inherited From	Initial Value	Specified In
spacecraft-tracking-mo-id	—	by CM	CREATE.inv
trajectory-file-id	—	= undefined	No
time-offset	—	= 0	No
tracking-state	—	= waiting	No
tracking-time	—	= 0	as needed
done-time	—	by CM	as needed
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

5.1.6 ACTIONS—AntennaPointing MO CLASS

A **new trajectory** action shall force the resource to immediately abandon tracking with its current trajectory prediction and switch to the new trajectory prediction file specified in the *trajectory-file-id* attribute, if the new file is valid. No specific *action-parameters* are associated with this action.

5.1.7 STATE-RELATED BEHAVIOR—AntennaPointing MO CLASS

5.1.7.1 General

See figure 5-1 for the antennaPointing state transition diagram. Since antennaPointing managed objects are contained by servicePackage managed objects, the first two states of the definition phase, *waiting* and *under validation*, are the same as for the ServicePackage managed object class. See 5.17.4 and 5.17.7 for a discussion of definition phase states and state transitions associated with the servicePackage managed object and its contained objects. We begin our discussion with the *validated* state.

NOTES

1 GET operation requests are accepted in all states.

2 No general rules are defined in this subsection for the acceptance or rejection of SET operation requests. They are state- and attribute-dependent and hence are part of the attributes description.

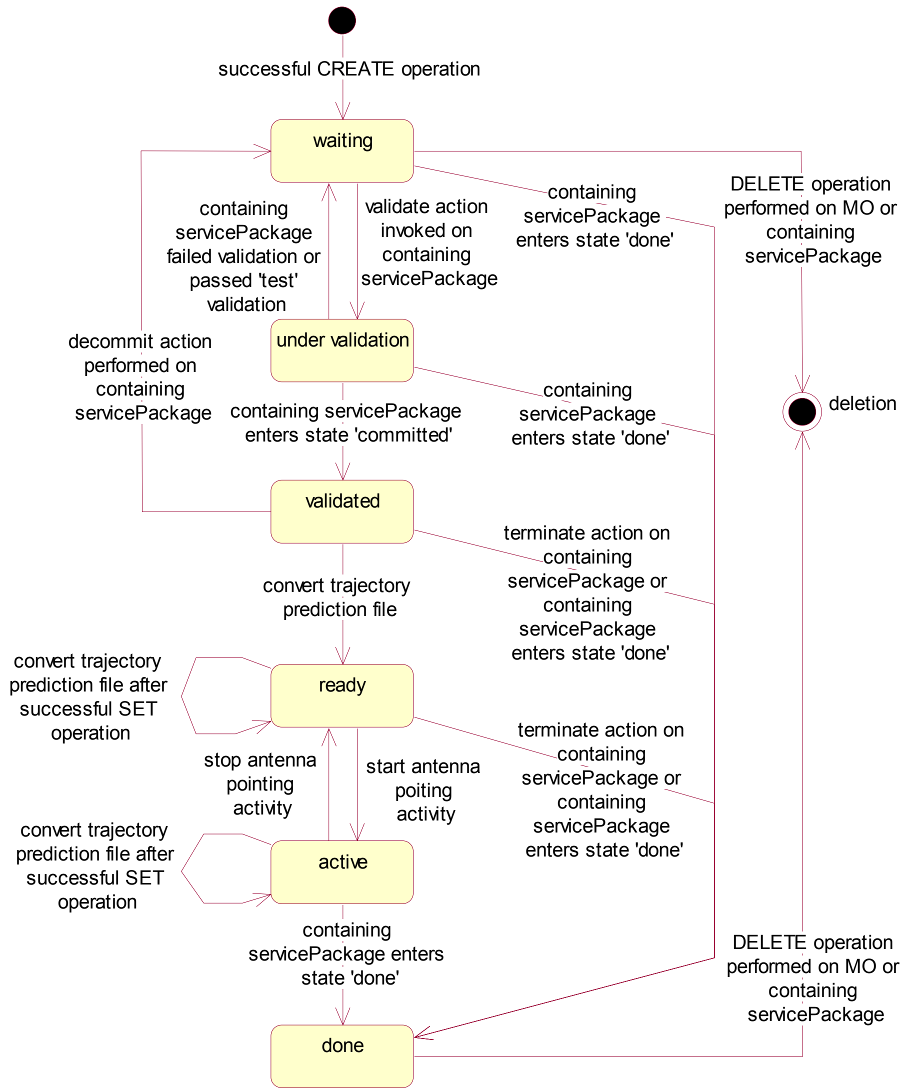


Figure 5-1: AntennaPointing State Transition Diagram

5.1.7.2 VALIDATED State

5.1.7.2.1 Conditions for the antennaPointing managed object entering the state *validated*

When the containing servicePackage managed object passes validation (the servicePackage validation process depends on a valid trajectory prediction file being made available via the attribute *trajectory-file-id*).

5.1.7.2.2 Actions and operations accepted and rejected by the antennaPointing managed object in the state *validated*

a) Rejects a DELETE invocation.

b) Rejects a *new trajectory* action invocation.

5.1.7.2.3 Actions and operations on containing managed objects that affect the antennaPointing managed object when in the state *validated*

a) *Decommit* action performed on the containing servicePackage managed object causes the antennaPointing managed object to transition to state *waiting*.

b) *Terminate* action performed on the containing servicePackage managed object causes the antennaPointing managed object to transition to state *done*.

c) *Abort* action invoked on the containing servicePackage managed object causes the antennaPointing managed object to transition to state *done*.

5.1.7.3 READY State

5.1.7.3.1 Conditions for the antennaPointing managed object entering the state *ready*

The antennaPointing managed object shall enter the state *ready* when the trajectory prediction file made available via the attribute *trajectory-file-id* has been converted into a format that the Complex's tracking resources can use. The file is ready for use.

NOTE – After a successful SET operation on the attribute, *trajectory-file-id*, the new trajectory prediction file will have to be converted into a format that the Complex's tracking resources can use.

5.1.7.3.2 Actions and operations accepted and rejected by the antennaPointing managed object in the state *ready*

a) A DELETE invocation shall be rejected.

b) A *new trajectory* action invocation shall be rejected.

5.1.7.3.3 Actions and operations on containing managed objects that affect the antennaPointing managed object when in the state *ready*

a) A *terminate* action performed on the containing servicePackage managed object shall cause the antennaPointing managed object to transition to state *done*.

b) An *abort* action performed on the containing servicePackage managed object shall cause the antennaPointing managed object to transition to state *done*.

5.1.7.4 ACTIVE State

5.1.7.4.1 Conditions for the antennaPointing managed object entering the state *active*

The antennaPointing managed object shall enter the state *active* at the start of an antenna-pointing activity when the managed object is in the state *ready*.

5.1.7.4.2 Actions and operations accepted and rejected by the antennaPointing managed object in the state *active*:

a) A DELETE invocation shall be rejected.

b) A *new trajectory* action invocation shall be accepted if the new trajectory file is valid.

5.1.7.4.3 Actions and operations on containing managed objects that affect the antennaPointing managed object when in the state *active*

An *abort* action performed on the containing servicePackage managed object shall cause the antennaPointing managed object to transition to state *done*.

5.1.7.5 DONE State

5.1.7.5.1 Conditions for the antennaPointing managed object entering the state *done*

a) When the containing servicePackage enters state *done.*

b) If the antennaPointing managed object is in state *validated* or *ready,* and a *terminate* action is performed on the containing servicePackage managed object.

5.1.7.5.2 Actions and operations accepted and rejected by the antennaPointing managed object in the state *done*

a) A DELETE invocation shall be accepted.

b) A *new trajectory* action invocation shall be rejected.

5.1.7.5.3 Actions and operations on containing managed objects that affect the antennaPointing managed object when in the state *done*

A DELETE operation performed on the containing servicePackage managed object shall cause the antennaPointing managed object to be deleted.

5.1.8 NOTIFICATIONS—AntennaPointing MO CLASS

An antennaPointing managed object issues the following state transition notifications:

a) **Invalid trajectory file.** The managed object has rejected a *new trajectory* action invocation because the trajectory file was invalid. An *invalid trajectory file* notification contains the additional information:

   1) *invoke-id.* Identification of the action that caused the generation of this notification.

   2) *trajectory-file-id*. The operationally applicable trajectory prediction file in the trajectory filestore.

   3) *diagnostic*. The reason for failure. Possible values for this parameter are:

i) 'File type not supported'. SLE Complex Management does not support the file type supplied.

ii) 'File cannot be found'. The file identified in the *trajectory-file-path* attribute could not be found.

iii) 'File is corrupted'. The new trajectory file could not be read.

iv) 'Orbit cannot be seen'. The new trajectory file supplies information on an orbit that cannot be seen by any of the Complex antennas for the SpaceLinkCarrierPackage sessions requested.

v) 'Inconsistent time intervals'. The new trajectory file contains information for time intervals inconsistent with the SpaceLinkCarrierPackage sessions and the service provision periods.

b) **Start antenna-pointing activity.** The managed object has transitioned from the state *ready* to *active.*

c) **Stop antenna-pointing activity.** The managed object has transitioned from the state *active* to *ready*.

5.1.9 VALIDATION ASPECTS—AntennaPointing MO CLASS

If the Complex is to perform space link acquisition, but a valid spacecraft trajectory prediction file is not identified when the servicePackage managed object that contains the antennaPointing managed object is validated, a 'completeness defect' is added to the appropriate servicePackage managed object's set of *defect-records*. The antennaPointing managed object shall remain in the *waiting* state.

NOTE – Presence of an antennaPointing managed object does not necessarily imply that a trajectory prediction information file is present.

5.2 AuthenticationInformationBase MANAGED OBJECT CLASS

5.2.1 PURPOSE

This concrete managed object class defines management information that SLE Utilization Management and SLE Complex Management must share in order to support the authentication used to secure the management and transfer service interactions between the MDOS and the SLE Complex. This managed object contains the various Authentication managed objects associated with the serviceAgreement managed object that contains it. This managed object class is specified in reference [12].

5.2.2 INHERITANCE—AuthenticationInformationBase MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

5.3 ChannelStored MANAGED OBJECT CLASS

5.3.1 PURPOSE

5.3.1.1 General

This abstract class encapsulates dynamic behavior and basic attributes of the xxx-st managed object classes, where xxx is an abbreviation of the name of a transfer service to be stored.

a) Data transfer may take place either before or after a Space Link session, depending on the direction of data flow. Forward data may be transferred and stored before a Space Link session so that it can be transmitted to the spacecraft later, while the Space Link is active. Return data may be collected during a Space Link session and transferred later as an offline service.

b) A channelStored managed object specifies basic properties of a storage service for SLE data channels. A storage service is a prerequisite for providing offline SLE transfer services.

c) Prerequisite for a storage service is the existence of a suitable Complex-internal storage resource where the SLE-SDUs carried in an SLE data channel can be deposited for subsequent retrieval.

NOTE – A dataStore managed object represents a storage resource.

d) Storage service is a two-step process:

   1) Ingest. This is the acquisition of SLE data channels and their deposit in a storage resource's partition.

      Ingest may take place in successive steps belonging to several distinct service packages. In each step, newly acquired SLE data channels can be added to the partition.

   2) Retrieval. The extraction of selected SLE data channels from the partition where they were deposited.

      Retrieval may be performed at any time—during or after ingest. A channelStored managed object with the 'retrieve' access mode specifies each retrieval from a given partition.

5.3.1.2 Perception by SLE Utilization Management

SLE Utilization Management shall be responsible for:

a) creating a storage partition on the referenced data storage resource;

b) harmonizing the storage period with the availability of the referenced storage resource;

c) harmonizing the lifetime of the partition with the mission's retrieval requirement;

d) directing the specified data channel to the storage resource;

e) complying with physical or policy restrictions on the use of storage at the SLE Complex (see 5.4.1.2 and 5.4.1.3).

5.3.1.3 Perception by SLE Complex Management

SLE Complex Management shall be responsible for:

a) relating the channelStored managed object to the appropriate dataStore resource partition;

b) providing the necessary resources for ingest and retrieval;

c) linking the ingest and retrieval resources to the dataStore resource partition;

d) linking the ingest and retrieval resources to the appropriate production or service provision resources;

e) activating and deactivating these resources as required by the ingest and retrieval windows.

5.3.2 INHERITANCE—ChannelStored MO CLASS

This managed object is derived from and inherits the properties of the SLEManagementEntity managed object class.

5.3.3 OBJECTS CONTAINED—ChannelStored MO CLASS

This managed object class is an abstract class; thus, it cannot be instantiated and containment relationships do not apply.

5.3.4 STATES—ChannelStored MO CLASS

a) In the definition phase, a channelStored managed object shall exist in one of the following states:

   1) **Waiting.** The managed object is created and may be edited or submitted to the validation process.

   2) **Under Validation.** The servicePackage managed object is undergoing validation. The managed object is static until validation is complete.

   3) **Validated.** The managed object is confirmed by the validation of the servicePackage managed object that contains it. All pertinent Complex-internal resources are allocated to the managed object.

b) In the utilization phase, a channelStored managed object shall exist in one of the following states:

1) **Configured.** The start of the ingest- or retrieval-window has been reached and a successful association between the managed object and the resource has been established and the resource has been successfully configured according to the managed object's attribute values.

2) **Unavailable.** Service provision is not possible for unexpected Complex-internal reasons.

3) **Completed.** The end of the ingest- or retrieval-window has been reached. Complex resources have been released.

4) **Done**. All contained managed objects have transitioned to *done*.

5.3.5 ATTRIBUTES—ChannelStored MO CLASS

This managed object class shall have the following specific attributes:

a) **data-store-id**. The DN of the dataStore managed object that supports the channelStored managed object.

   1) Data type/values: DN.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

b) **partition-id**. The partition where the SLE data channels of interest are held.

   1) Data type/values: ASCII character string of 256 or fewer characters

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **access-mode**. The access mode for the designated partition.

   1) Data type/values:

      i) 'append' if write-only access to the designated partition is required;

      ii) 'retrieve' if read-only access to the designated partition is required;

      iii) 'append-retrieve' if simultaneous read and write access is required.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting* .

d) **ingest-window-start-time**. The earliest time at which SLE data channels may be written into the partition.

   1) Data type/values: CCSDS time code.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

3) Dependencies: This attribute is mandatory if *access-mode* is 'append' or 'append-retrieve'.

e) **ingest-window-stop-time**. The latest time after which no SLE data channels may be written into the partition.

1) Data type/values: CCSDS time code

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

3) Dependencies: This attribute is mandatory if *access-mode* is 'append' or 'append-retrieve'.

f) **ingest-time**. Accumulated ingest time. The difference (in seconds) between the current time and *ingest-window-start-time*.

1) Data type/values: Integer. The initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

3) Dependencies: Must not exceed the difference between *ingest-window-stop-time* and *ingest-window-start-time*.

g) **retrieval-window-start-time**. The earliest time at which SLE data channels may be retrieved from the partition.

1) Data type/values: CCSDS time code

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

3) Dependencies: This attribute is mandatory if *access-mode* is 'retrieve' or 'append-retrieve'.

h) **retrieval-window-stop-time**. The latest time after which no data can be retrieved from the partition.

1) Data type/values: CCSDS time code

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

3) Dependencies: This attribute is mandatory if *access-mode* is 'retrieve' or 'append-retrieve'.

i) **unavailable-time**. The accumulated time the managed object is in the state *unavailable* until reporting time.

   1) Data type/values: seconds.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

   3) Dependencies: Must not exceed the difference between *ingest-* or *retrieval-window-stop-time* and *ingest-* or *retrieval-window-start-time.*

j) **provision-time**. The accumulated time (in seconds) the managed object is in the state *configured* until reporting time.

   1) Data type/values: Integer.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

   3) Dependencies: Must not exceed the difference between *ingest-* or *retrieval-window-stop-time* and *ingest-* or *retrieval-window-start-time.*

k) **number-SLE-SDUs-stored.** The number of SLE-PDUs in store at reporting time.

   1) Data type/values: integer.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

      i) SET shall be disabled in both the definition and utilization phases;

      ii) GET shall be disabled in the definition phase.

l) **channel-stored-state**. The current state of the channelStored managed object.

   1) Data type/values: See 5.3.4.

   2) Modifications: SLE Complex Management may modify this attribute at any time, as follows: SET shall be disabled in both the definition and utilization phases.

n) **done-time**. The time when the managed object entered the state *done.*

   1) Data type/values: CCSDS time code.

2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

3) Dependencies: Must be between *ingest-* or *retrieval-window-start-time* and *ingest-* or *retrieval-window-stop-time*.

o) **partition-data-units**. The number of data units in the partition.

1) Data type/values: Unsigned integer. Initial value shall be ‘0’.

2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

Refer to table 5-2 for the channelStored managed object attribute value summary.

Table 5-2: Attribute Value Summary—ChannelStored Managed Object

Attribute	Inherited From	Initial Value	Specified in
data-store-id	—	by UM	CREATE.inv
partition-id	—	by UM	CREATE.inv
access-mode	—	by UM	CREATE.inv
ingest-window-start-time	—	by UM	CREATE.inv
ingest-window-stop-time	—	by UM	CREATE.inv
ingest-time	—	= 0	CREATE.rtn
retrieval-window-start-time	—	by UM	CREATE.inv
retrieval-window-stop-time	—	by UM	CREATE.inv
unavailable-time	—	= 0	CREATE.rtn
provision-time	—	= 0	CREATE.rtn
number-SLE-SDUs-stored	—	= 0	CREATE.rtn
channel-stored-state	—	= waiting	CREATE.inv
done-time	—	by CM	when MO enters *done*
partition-data-units	—	= 0	as needed
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

5.3.6 ACTIONS—ChannelStored MO CLASS

5.3.6.1 Abort Action

a) An **abort** action inhibits the related storage activities even after a successful validation.

b) No specific *action-parameters* are associated with this action.

5.3.6.2 Terminate Action

a) A **terminate** action deactivates the channelStored managed object immediately after completing an ongoing ingest or retrieval of an SLE-PDU.

b) No specific *action-parameters* are associated with this action.

5.3.7 STATE-RELATED BEHAVIOR—ChannelStored MO CLASS

5.3.7.1 General

See figure 5-2 for the channelStored state transition diagram. Since channelStored managed objects are contained by servicePackage managed objects, the first two states of the definition phase, *waiting* and *under validation*, are the same as for the ServicePackage managed object class. Refer to 5.17.4 and 5.17.7 for a discussion of definition phase states and state transitions associated with the servicePackage managed object and its contained objects. We begin our discussion with the *validated* state.

NOTES

1 GET operation requests are accepted in all states.

2 No general rules are defined in this subsection for the acceptance or rejection of SET operation requests. They are state- and attribute-dependent and hence are part of the attributes description.

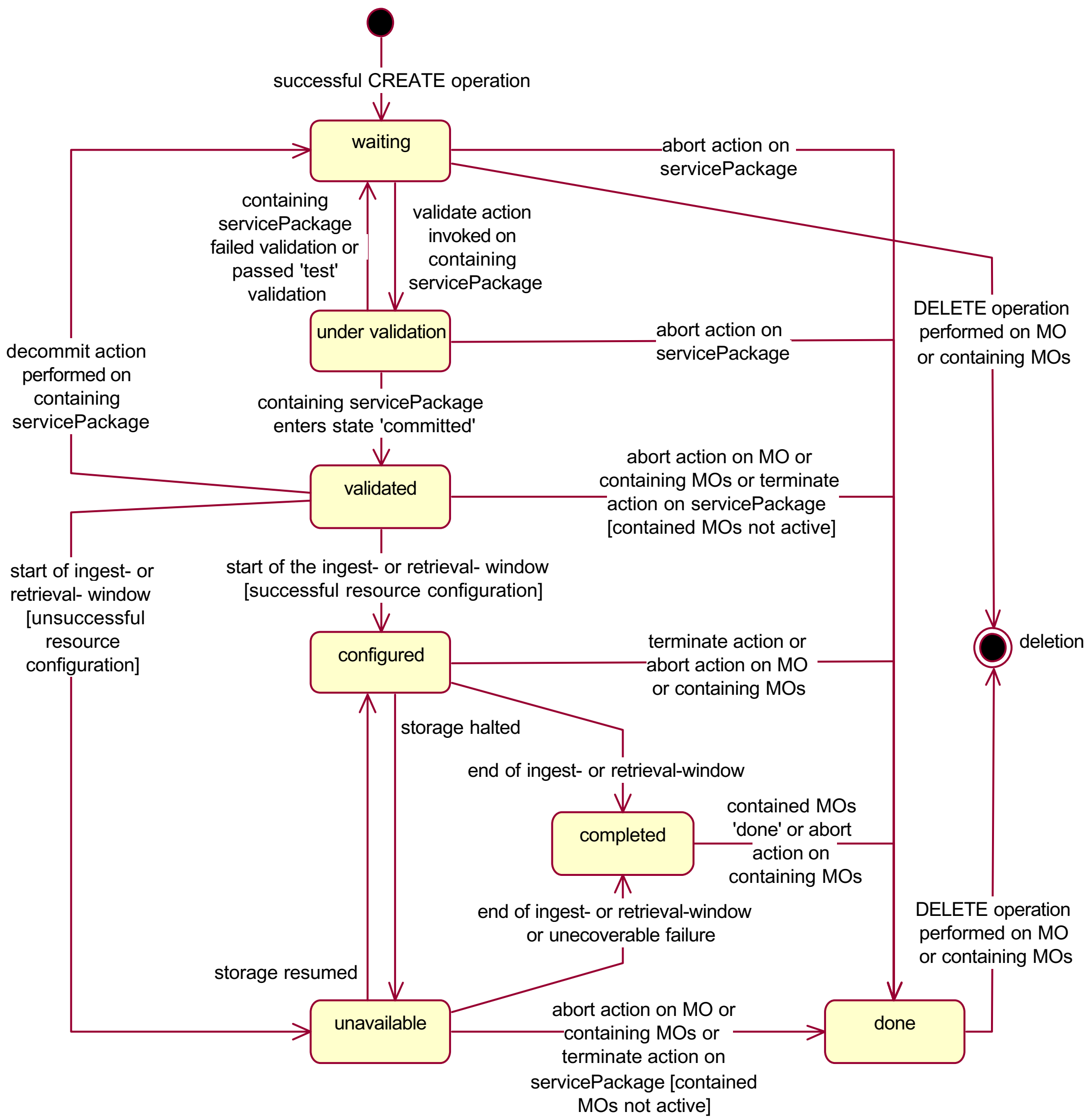


Figure 5-2: ChannelStored State Transition Diagram

5.3.7.2 VALIDATED State

5.3.7.2.1 General

When in the state *validated*, the channelStored managed object waits for the start of the ingest- or retrieval-window.

5.3.7.2.2 Conditions for the channelStored managed object entering the state validated

If it passed the validation process and the servicePackage managed object that contains it is *committed*, then the channelStored managed object can enter the state *validated*.

5.3.7.2.3 Actions and operations accepted and rejected by the channelStored managed object in the state *validated*

a) Rejects a DELETE invocation.

b) Rejects a CREATE invocation for contained managed objects.

c) Accepts an *abort* action invocation.

d) Rejects a *terminate* action invocation.

5.3.7.2.4 Actions and operations on containing managed objects that affect the channelStored managed object when in the state *validated*

a) *Decommit* action performed on the containing servicePackage managed object causes the channelStored managed object to transition to state *waiting*.

b) *Terminate* action performed on the containing servicePackage managed object causes the channelStored managed object to transition to state *done* if none of its contained managed objects is active (the states in which managed objects are considered ‘active’ are listed in 4.10).

c) *Abort* action performed on containing managed objects causes a state transition to *done*.

5.3.7.3 CONFIGURED State

5.3.7.3.1 General

When in the state *configured*, the channelStored managed object supports ingest or retrieval operations as specified by its attributes.

5.3.7.3.2 Conditions for the channelStored managed object entering the state *configured*

a) At the start of the ingest- or retrieval- window, if SLE Complex Management has established an association between the managed object and the resource, and the resource has been successfully configured according to the managed object’s attribute values.

b) If storage service is resumed when channelStored managed object is in state *unavailable.*

5.3.7.3.3 Actions and operations accepted and rejected by the channelStored managed object in the state *configured*

a) Rejects a DELETE invocation.

b) Rejects a CREATE invocation for contained managed objects.

c) Rejects an *abort* action invocation.

d) Accepts a *terminate* action invocation.

5.3.7.3.4 Actions and operations on containing managed objects that affect the channelStored managed object when in the state *configured*

Abort action performed on the containing managed object causes a state transition to *done*.

5.3.7.4 UNAVAILABLE State

5.3.7.4.1 General

When in the state *unavailable*, the channelStored managed object waits for service to be resumed.

5.3.7.4.2 Conditions for the channelStored managed object entering the state *unavailable*

If managed object fails to establish association with or fails to configure local service-providing resources.

5.3.7.4.3 Actions and operations accepted and rejected by the channelStored managed object in the state *unavailable*

a) Rejects a DELETE invocation.

b) Rejects a CREATE invocation for contained managed objects.

c) Accepts an *abort* action invocation.

d) Rejects a *terminate* action invocation.

5.3.7.4.4 Actions and operations on containing managed objects that affect the channelStored managed object when in the state *unavailable*

a) *Terminate* action performed on the containing servicePackage managed object causes the channelStored managed object to transition to state *done* if none of its contained managed objects is active (the states in which managed objects are considered 'active' are listed in 4.10).

b) *Abort* action performed on the containing managed objects causes a state transition to *done*.

5.3.7.5 COMPLETED State

5.3.7.5.1 General

When in the state *completed*, the channelStored managed object is dissociated from its resources.

5.3.7.5.2 Conditions for the channelStored managed object entering the state *completed*

a) At end of ingest- and retrieval- window.

b) If recovery from corruption cannot be achieved.

5.3.7.5.3 Actions and operations accepted and rejected by the channelStored managed object in the state *completed*

a) Rejects a DELETE invocation.

b) Rejects a CREATE invocation for contained managed objects.

c) Rejects an *abort* action invocation.

d) Rejects a *terminate* action invocation.

5.3.7.5.4 Actions and operations on containing managed objects that affect the channelStored managed object when in the state *completed*

Abort action performed on the containing managed objects causes a state transition to *done*.

5.3.7.6 DONE State

5.3.7.6.1 Conditions for the channelStored managed object entering the state *done*

a) If all of contained MOs have entered state *done*, when managed object is in state *completed*.

b) Upon a *terminate* action.

c) If a *terminate* action is performed on the containing servicePackage managed object when the channelStored managed object is in state *validated* or *unavailable* and none of the contained managed objects is active (the states in which managed objects are considered 'active' are listed in 4.10).

d) When an *abort* action is performed on the managed object or on the containing managed objects.

5.3.7.6.2 Actions and operations accepted and rejected by the channelStored managed object in the state *done*:

a) Accepts a DELETE invocation.

b) Rejects a CREATE invocation for contained managed objects.

c) Rejects an *abort* action invocation.

d) Rejects a *terminate* action invocation.

5.3.7.6.3 Actions and operations on containing managed objects that affect the channelStored managed object when in the state *done*

DELETE operation performed on the containing managed object causes the channelStored managed object to be deleted.

5.3.8 NOTIFICATIONS—ChannelStored MO CLASS

A channelStored managed object issues the following state transition notifications:

a) **Start of storage period.** The managed object has transitioned from the state *validated* to *configured*.

b) **End of storage period.** The managed object has transitioned to the state *completed* because the end of the ingest- or retrieval-window period has been reached.

c) **Service halted.** The managed object has transitioned from the state *configured* to *unavailable*. A *service halted* notification contains the additional information:

   Reason for interruption. An implementation-specific value within a specific Complex. The value is provided by SLE Complex Management. It is a character string.

d) **Service resumed.** The managed object has transitioned from the state *unavailable* to *configured*.

e) **Unrecoverable failure.** The managed object has transitioned from the state *unavailable* to *done*.

5.3.9 VALIDATION ASPECTS—ChannelStored MO CLASS

a) The referenced dataStore managed object must be contained in the serviceAgreement managed object. If not, a 'completeness defect' entry 'missing data store' shall be added to the set of *defect-records* in the servicePackage managed object that contains the channelStored managed object.

b) The *ingest-window-start time* and *ingest-window-stop-time* must be within the lifetime of the referenced dataStore managed object if this attribute is present. If this is not the case, a 'consistency defect' entry 'time mismatch' shall be added to the set of *defect-records* in the servicePackage managed object that contains the channelStored managed object. The validation shall fail.

c) The *retrieval-window-start-time* and *retrieval-window-stop-time* must be within the lifetime of the referenced dataStore managed object, if this attribute is present. It must not begin before the start of the *ingest-window-start-time* attribute. If this is not the case, a consistency defect entry 'time mismatch' shall be added to the set of *defect-records* in the servicePackage managed object that contains the channelStored managed object. The validation shall fail.

d) If the *access-mode* is 'append' or 'append-retrieve' and the designated partition does not yet exist, the SLE Complex Management shall establish a new partition on the designated data storage resource. The new partition shall be referenced with the value given to *partition-id*.

e) If the *access-mode* is 'retrieve' and the designated partition does not exist, the validation shall fail and a 'consistency defect' entry 'partition missing' shall be added to the set of *defect-records* in the servicePackage managed object that contains the channelStored managed object.

5.4 DataStore MANAGED OBJECT CLASS

5.4.1 PURPOSE

5.4.1.1 General

a) A dataStore managed object represents a Complex's storage resource and is dedicated to a given SLE Utilization Management.

b) A storage resource is used to store SLE data channels as they become available from a Complex's transfer service ports or from Complex-internal production resources.

c) The data store may be divided into several partitions. A partition can accommodate only a single type of SLE data channel.

NOTES

1 A Complex may store and deliver return data to customers via media not covered by this draft Recommendation, such as physical media sent by surface mail to various destinations. However, in this case, the requirements for data acquisition and storage will be covered by a dataStore managed object.

2 The need for a catalog that reflects the contents of a dataStore is for further study. Information that may be useful includes missing-data flags and observation time periods from CCSDS compliant Packet Secondary Headers.

5.4.1.2 Perception by SLE Utilization Management

a) A data store's lifetime may cover several service packages' lifetimes. Hence, it is possible to accumulate data in it (e.g., telemetry) that is incrementally acquired during several Space Link sessions, and provide it as a single data set for eventual retrieval.

NOTES

1 In order to optimize the usage patterns and cost of storage resources, SLE Utilization Management may establish several dataStore managed objects with different sizes and lifetimes.

2 SLE Utilization Management may need to establish multiple dataStore managed objects to reserve storage at different physical locations within an SLE Complex (e.g., at two geographically separate ground stations under a single SLE Complex Management).

b) SLE Utilization Management shall be notified if the storage resource is filled to a specified level.

c) To adjust the overall storage availability to mission needs, SLE Utilization Management may request a reduction or extension of the storage resource's lifetime and/or size.

d) SLE Utilization Management may divide a storage resource into several named partitions.

e) SLE Utilization Management shall be able to establish and access a partition using a channelStored managed object.

f) A given partition shall accommodate SLE channels of a single type.

g) A given partition may be fed incrementally during the lifetime of several distinct servicePackage managed objects.

h) SLE data channels may be retrieved from a given partition during the lifetime of several distinct servicePackage managed objects.

NOTES

1 Retrievals are specified by means of 'service provided' (xxx-ts-p, where xxx identifies a particular transfer service) managed objects contained by a suitable channelStored managed object.

2 Provision of standardized retrieval keys (e.g., time, SLE data channel type, observation period, etc.) is for further study. In particular, useful retrieval keys for forward data must be investigated.

i) SLE Utilization Management may clear a partition if it is not in use (i.e., at the time of the *clear* action invocation, the partition is used for neither storage nor retrieval).

j) SLE Utilization Management may investigate the size and current usage of an individual partition.

5.4.1.3 Perception by SLE Complex Management

a) The total amount of storage to be made available to an SLE Utilization Management must be negotiated and agreed upon before the Service Agreement becomes *applicable*.

b) SLE Complex Management may require creation of separate data stores at separate physical locations within the Complex. SLE Complex Management may require creation of separate data stores for data with different characteristics (e.g., transfer rate, storage time, security requirements).

c) The storage resource shall support:

1) establishment and release of partitions;

2) one feed at a time into a given partition;

3) parallel feeding into several distinct partitions;

4) parallel retrievals from a given partition;

5) parallel retrievals from several distinct partitions;

6) feed and retrieve operations in parallel on a given partition;

7) persistence of the data stored over several service package lifetimes, if required.

5.4.1.4 Creation Aspects—DataStore MO Class

a) Creation of a dataStore managed object may be requested at any time during the Service Agreement period.

b) Creation of a dataStore managed object is subject to the following preconditions:

1) the number of already existing dataStore managed objects must be below the maximum number of dataStore managed objects specified in the serviceAgreement managed object;

2) the lifetime requested must fall into the Service Agreement period;

3) during the lifetime of the new dataStore managed object, the accumulated sizes of co-existing dataStore managed objects plus the additional size requested for the new dataStore managed object must remain below the total size-allowance for all dataStore managed objects specified in the serviceAgreement managed object.

5.4.1.5 Security Aspects—DataStore MO Class

A mission has access only to storage resources associated with dataStore managed objects contained by its own serviceAgreement managed object.

5.4.1.6 Deletion Aspects—DataStore MO Class

Deletion of a dataStore managed object causes the release of the storage resource associated with this dataStore managed object and, consequently, the loss of all data acquired.

5.4.2 INHERITANCE—DataStore MO CLASS

This managed object is derived from and inherits the properties of the SLEManagementEntity managed object class.

5.4.3 OBJECTS CONTAINED—DataStore MO CLASS

An instance of this concrete managed object class contains no other managed objects.

5.4.4 STATES—DataStore MO CLASS

A dataStore managed object shall exist in one of four states:

a) **Validated.** The dataStore managed object has been created, but is not yet associated with a storage resource.

b) **Configured.** All pertinent Complex-internal resources have been allocated to the dataStore managed object.

c) **Unavailable.** Data storage is not possible for Complex-internal reasons.

d) **Done.** The dataStore managed object is dissociated from the storage resource.

5.4.5 ATTRIBUTES—DataStore MO CLASS

This managed object class has the following specific attributes:

a) **data-store-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (see annex C).

   1) Data type/values: Character string.

   2) Modifications: SET shall be disabled in all states.

b) **lifetime-start.** The start of the period during which the storage resource is operational. This attribute value is specified by SLE Utilization Management in the CREATE invocation.

   1) Data type/values: CCSDS time code.

   2) Modifications: SET shall be enabled when the managed object is in the state *validated*.

c) **lifetime-stop.** The end of the period during which the storage resource is operational. This attribute value is specified by SLE Utilization Management in the CREATE invocation.

   1) Data type/values: CCSDS time code.

   2) Modifications: SET shall be enabled in the states *validated, configured* and *unavailable.*

      i) If SLE Complex Management decides that the data storage resource cannot be allocated for an extended period, a SET error shall be returned with *diagnostic* ‘inadequate resource’, and the old value shall be retained.

ii) If the data storage resource can remain allocated for an extended period, the modification shall succeed.

d) **total-size.** The total storage size available from the storage resource, in multiples of megabytes. The initial value of this attribute is specified by SLE Utilization Management in the CREATE invocation. SLE Utilization Management is responsible for the usage of this storage volume.

1) Data type/values: Integer.

2) Modifications: SET shall be enabled in the states *validated, configured* and *unavailable.*

i) If the size of the data store resource cannot be modified, the SET operation shall fail with *diagnostic* ‘inadequate resource’ and the old value shall be retained.

ii) A request from SLE Utilization Management to reduce the total size shall succeed only if the data stored can be accommodated in the reduced total size; if this is not the case, the request shall fail with *diagnostic* ‘inadequate resource’.

e) **alarm-threshold.** The threshold value after which a *threshold exceeded* notification is issued, if during ingest the storage resource is filled beyond the threshold. This attribute value is specified (as a percentage of the *total-size* attribute) by SLE Utilization Management in the CREATE invocation.

1) Data type/values: Integer in the range 0 to 100.

2) Modifications: SET shall be enabled in the states *validated, configured* and *unavailable.*

f) **max-rate-in.** The maximum input rate (in bits per second) that the storage resource must handle. This attribute value is specified by SLE Utilization Management in the CREATE invocation.

1) Data type/values: Integer.

2) Modifications: SET shall be enabled when the managed object is in the state *validated*.

g) **data-store-state.** The current state of the dataStore managed object set by SLE Complex Management.

1) Data type/values: See 5.4.4.

2) Modifications: SET shall be disabled in all states.

h) **percent-resource-filled**. The percentage of the storage resource that is currently filled.

1) Data type/values: Integer in the range 0 to 100.

2) Modifications:

   i) SET shall be disabled in all states.

   ii) GET shall be enabled when the managed object is in the states *configured, unavailable,* and *done*.

i) **active-partition-ids**. A set of currently active *partition-ids*.

1) Data type/values: Initial value shall be 'empty'.

2) Modifications:

   i) SET shall be disabled in all states.

   ii) GET shall be enabled when the managed object is in the states *configured, unavailable,* and *done*.

Refer to table 5-3 for the dataStore managed object attribute value summary.

Table 5-3: Attribute Value Summary—DataStore Managed Object

Attribute	Inherited From	Initial Value	Specified in
data-store-mo-id	—	by CM	CREATE.inv
lifetime-start	—	by UM	CREATE.inv
lifetime-stop	—	by UM	CREATE.inv
total-size	—	by UM	CREATE.inv
alarm-threshold	—	by UM	CREATE.inv
max-rate-in	—	by UM	CREATE.inv
data-store-state	—	= waiting	No
percent-resource-filled	—	= empty	as needed
active-partition-ids	—	by CM	as needed
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

5.4.6 ACTIONS—DataStore MO CLASS

5.4.6.1 List Partitions Action

a) A **list partitions** action provides information about each partition on the storage resource.

b) A *list partitions* action has no *action-parameters*.

c) A *list partitions* action returns the following information:

1) **number of partitions**. If no partition exists when the action is invoked, the number shall be 0 and no partition information shall appear in the list; otherwise, the number of the existing partitions shall be returned and the respective partition information shall be provided.

2) **partition information**: For each partition that exists when the action is invoked, the following information shall be returned:

   i) **partition id.** The partition's identification as specified in the channelStored managed object that originally caused the partition to be created. This item is an ASCII character string of 256 or fewer characters.

   ii) **partition size.** The current size of the partition in integral mega bytes.

   iii) **partition usage.** Partition usage shall have the value 'idle' if the partition is not associated with a channelStored managed object; it shall have the value 'active' if the partition is associated with a channelStored managed object.

5.4.6.2 Clear Partition Action

a) A **clear partition** action removes the specified partition from the storage resource and releases the storage that was occupied.

b) A *clear partition* action invocation has the *action-parameter partition-id,* which specifies the partition to be removed. If an active channelStored managed object is accessing this partition, or the partition referenced does not exist, the *clear partition* ACTION operation shall fail with *diagnostic* 'partition not available'.

5.4.7 STATE-RELATED BEHAVIOR—DataStore MO CLASS

5.4.7.1 General

See figure 5-3 for the dataStore state transition diagram.

NOTES

1 GET operation requests are accepted in all states.

2 No general rules are defined in this subsection for the acceptance or rejection of SET operation requests. They are state- and attribute-dependent and hence are part of the attributes description.

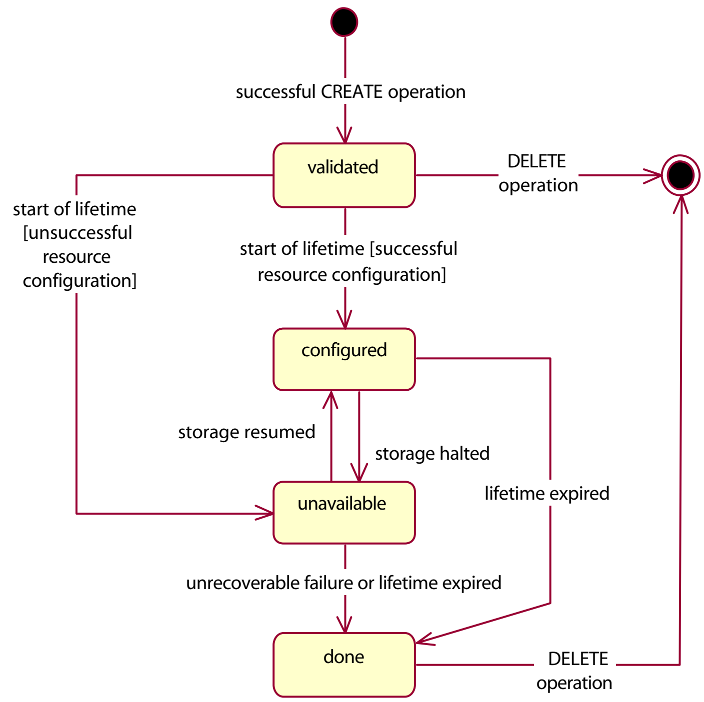


Figure 5-3: DataStore State Transition Diagram

5.4.7.2 VALIDATED State

5.4.7.2.1 General

When in the state *validated*, the dataStore managed object does not support channelStored managed objects' feed or retrieval activities.

5.4.7.2.2 Conditions for the dataStore managed object entering the state *validated*

A dataStore managed object enters the state *validated* at creation time.

NOTE – Comprehensive checking at creation time ensures that a request for an unfeasible data store always fails. Therefore, the initial state of this managed object is *validated.*

5.4.7.2.3 Actions and operations accepted and rejected by the dataStore managed object in the state *validated*

a) Accepts a DELETE invocation.

b) Accepts a *list partitions* action invocation.

c) Rejects a *clear partition* action invocation.

5.4.7.3 CONFIGURED State

5.4.7.3.1 Conditions for the dataStore managed object entering the state *configured*

a) If *lifetime-start* is reached and the SLE Complex Management has established an association between the managed object and the resource, and the resource has been successfully configured.

b) If storage service is resumed when managed object is in state *unavailable*.

5.4.7.3.2 Actions and operations accepted and rejected by the dataStore managed object in the state *configured*

a) Rejects a DELETE invocation.

b) Accepts a *list partitions* action invocation.

c) Accepts a *clear partition* action invocation.

5.4.7.4 UNAVAILABLE State

5.4.7.4.1 Conditions for the dataStore managed object entering the state *unavailable*

a) If managed object fails to establish association with or fails to configure storage resources.

b) If storage resources become unavailable (e.g., due to corruption on the resource itself, or on the communication between resource and managed object).

5.4.7.4.2 Actions and operations accepted and rejected by the dataStore managed object in the state *active*

a) Rejects a DELETE invocation.

b) Rejects a *list partitions* action invocation.

c) Rejects a *clear partition* action invocation.

5.4.7.5 DONE State

5.4.7.5.1 Conditions for the dataStore managed object entering the state *done*

a) When *lifetime-stop* is reached.

b) If recovery from corruption cannot be achieved.

5.4.7.5.2 Actions and operations accepted and rejected by the dataStore managed object in the state *done*

a) Accepts a DELETE invocation.

b) Accepts a *list partitions* action invocation.

c) Rejects a *clear partition* action invocation.

5.4.8 NOTIFICATIONS—DataStore MO CLASS

a) A dataStore managed object issues the following state transition notifications:

   1) **Start of lifetime.** The managed object has transitioned from the state *validated* to *configured*.

   2) **Storage halted**. The managed object has transitioned from the state *configured* to *unavailable*. This notification contains the additional information:

      Reason for interruption. An implementation-specific value within a specific Complex. The value is provided by SLE Complex Management. It is a character string.

   3) **Storage resumed**. The managed object has transitioned from the state *unavailable* back to *configured*.

   4) **Unrecoverable failure.** The managed object has transitioned from the state *unavailable* to *done*.

   5) **Lifetime expired.** The managed object has transitioned from the state *configured* to *done*.

b) A dataStore managed object issues the following unique notification:

      Threshold exceeded. The storage resource has been filled beyond the threshold. The *threshold exceeded* notification contains the additional information:

         Alarm-threshold. The threshold value specified (as a percentage of the *total-size* attribute) by SLE Utilization Management in the CREATE invocation.

5.4.9 VALIDATION ASPECTS—DataStore MO CLASS

If a servicePackage managed object contains (on any level) one or more channelStored managed objects, the presence of a suitable dataStore managed object is mandatory. If the referenced dataStore managed object is missing, a ‘completeness defect’ shall be added to the appropriate servicePackage managed object’s set of *defect-records*.

5.5 EventControl MANAGED OBJECT CLASS

5.5.1 PURPOSE

An EventControl managed object serves as the container of managed objects associated with event handling.

5.5.2 INHERITANCE—EventControl MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

5.5.3 OBJECTS CONTAINED—EventControl MO CLASS

An instance of this managed object class shall contain instances of the following classes:

a) Zero, one, or more instances of the EventHandler managed object class. This class is specified in 5.6.

b) Zero, one, or more instances of the ProxyAction managed object class. This class is specified in 5.11.

c) Zero, one, or more instances of the ProxySet object class. This class is specified in 5.12.

d) Absence of both proxySet and proxyAction managed objects results in SLE Utilization Management only receiving an *event-notification* (see 5.6.8) that the event occurred. No further management action shall be taken.

5.5.4 STATES—EventControl MO CLASS

An EventControl managed object has no states.

5.5.5 ATTRIBUTES—EventControl MO CLASS

This managed object class shall have the following specific attribute: **ec-mo-id** (the RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class). (See annex C.)

1) Data type/values: Character string.

2) Modifications: SET shall be disabled in the definition and utilization phases.

Refer to table 5-4 for the eventControl managed object attribute value summary.

Table 5-4: Attribute Value Summary—EventControl Managed Object

Attribute	Inherited From	Initial Value	Specified In
ec-mo-id	—	by CM	CREATE.inv
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

5.5.6 ACTIONS—EventControl MO CLASS

An EventControl managed object performs no actions.

5.5.7 STATE-RELATED BEHAVIOR—EventControl MO CLASS

An EventControl managed object has no states.

5.5.8 NOTIFICATIONS—EventControl MO CLASS

An EventControl managed object issues no notifications.

5.5.9 VALIDATION ASPECTS—EventControl MO CLASS

No specific validation aspects are identified for this managed object.

5.6 EventHandler MANAGED OBJECT CLASS

5.6.1 PURPOSE

5.6.1.1 General

An eventHandler managed object acts as a proxy for SLE Utilization Management during provision of an SLE service package. It reacts to predefined events by performing operations on other managed objects in the SLE service package. This allows quick reaction to the event, even if there is no telecommunication association between SLE Utilization Management and SLE Complex Management.

a) At Service Agreement time, a list of events is identified for which special handling may be necessary. This list is contained in the Service Agreement. An eventHandler managed object is concerned with one, and only one, of these events.

b) If the event occurs the eventHandler is triggered, causing the *to-do-list* to be executed. On behalf of SLE Utilization Management, it may then invoke actions on other managed objects contained in the same servicePackage managed object. The *to-do-list* references one or more proxySet or proxyAction managed objects, each of which specifies a single proxy operation to be invoked.

NOTE – In the remainder of this EventHandler managed object definition subsection, the phrase 'execute the *to-do-list*' is used as an abbreviated way of saying 'invoke the proxy operations specified by the proxySet and proxyAction managed objects referenced by the *to-do-list*'.

c) An eventHandler may be configured to limit the number of times the *to-do-list* will be executed in the case where the event occurs more than once.

d) Regardless of whether the event occurred, SLE Utilization Management shall have the option of invoking an action to enforce the execution of the *to-do-list*.

5.6.1.2 Perception by SLE Utilization Management

SLE Utilization Management is responsible for:

a) Direct management actions performed on the exposed managed objects. However, SLE Utilization Management may choose to delegate indirect management actions to eventHandler managed objects.

NOTE – When a management association between SLE Utilization Management and SLE Complex Management is functioning properly, SLE Utilization Management may choose to disable an eventHandler managed object.

b) Creating multiple eventHandler managed objects if needed.

c) Ensuring that the target managed objects on which management operations are to be performed have been created beforehand.

d) Specifying the proxy operations appearing in the *to-do-list* of its eventHandler managed objects.

e) Enforcing the execution of an eventHandler managed object's *to-do-list* by means of the *enforce-to-do-list* action, if needed. This means that the *to-do-list* is executed regardless of whether the event occurred.

5.6.1.3 Perception by SLE Complex Management

SLE Complex Management shall be responsible for:

a) Constructing an event by observing the behavior of Complex-internal resources.

b) Associating a resource-event with an *event-identifier* as specified in the serviceAgreement managed object.

5.6.1.4 Deletion Aspects

Subsection 5.6.7 contains information about the states in which an eventHandler managed object can be deleted.

5.6.2 INHERITANCE—EventHandler MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

5.6.3 OBJECTS CONTAINED—EventHandler MO CLASS

An instance of this concrete managed object class contains no other managed objects.

5.6.4 STATES—EventHandler MO CLASS

a) In the definition phase, an eventHandler managed object shall exist in one of the following states:

   1) **Waiting.** The eventHandler managed object has been created and may be edited or submitted to the validation process.

2) **Under validation.** The servicePackage managed object is undergoing validation. The eventHandler managed object is static until validation is complete.

3) **Validated.** The eventHandler managed object is confirmed by the validation process of the servicePackage managed object that contains it.

b) In the utilization phase, an eventHandler managed object shall exist in one of the following states:

1) **Configured**. An association between the managed object and the resource has been established, and the resource has been successfully configured according to the managed object's attribute values.

2) **Enabled**. The eventHandler managed object will execute its *to-do-list* accordingly when the specified event occurs. While executing the *to-do-list*, the eventHandler managed object ignores re-occurrences of the specified event.

3) **Disabled**. The eventHandler managed object will ignore the specified event.

4) **Done**. The Complex resources are released. This state is equivalent to the state *done* of the containing servicePackage managed object.

5.6.5 ATTRIBUTES—EventHandler MO CLASS

5.6.5.1 General

This managed object class has the following specific attributes:

a) **event-handler-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (see annex C).

1) Data type/values: Character string.

2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **event-handler-state.** The current state of the eventHandler managed object.

1) Data type/values: See 5.6.4.

2) Modifications: SLE Complex Management may modify this value at any time. SET shall be disabled in both the definition and utilization phases.

c) **event-identifier.** An event at the Complex that triggers the eventHandler. The eventHandler will then execute the *to-do-list*. This event identifier must be selected from the *event-list* specified in the serviceAgreement managed object.

1) Data type/values: Integer.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

d) **initiating-trigger-delay.** The number of seconds that elapse between the time that the eventHandler is triggered and the time that the *to-do-list* is executed.

   1) Data type/values: Integer.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

e) **to-do-execution-mode.** How the *to-do-list* attribute will be executed.

   1) Data type/values:

      i) 'in sequence'. The proxy operations specified in the *to-do-list* attribute will be executed in the sequence in which they appear.

      ii) 'best effort'. No sequence is prescribed for the execution.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

f) **to-do-limitation.** The limitations applied to triggering eventHandler managed objects; it identifies the conditions that force the eventHandler into the *disabled* state relative to the execution of the *to-do-list*.

   1) Data type/values:

      i) 'once'. The *to-do-list* is to be executed only once. The eventHandler will be disabled for subsequent occurrences of the event.

      ii) 'forever'. The *to-do-list* is to be executed for all occurrences of the event until the eventHandler is disabled.

      iii) 'time window'. The *to-do-list* is to be executed for all occurrences of the event during the time window bounded by *to-do-start* and *to-do-stop*.

      iv) 'max trigger'. The *to-do-list* is to be executed for all occurrences of the event up to a maximum number of executions.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

g) **to-do-start.** The start time of the time window for executing the *to-do-list*.

   1) Data type/values: CCSDS time code.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

   3) Dependencies: This attribute has meaning only when the value of *to-do-limitation* equals 'time window'.

h) **to-do-stop**. The stop time of the time window for executing the *to-do-list*.

1) Data type/values: CCSDS time code.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

3) Dependencies: This attribute has meaning only when the value of *to-do-limitation* equals ‘time window’.

i) **failure-trigger-delay**. The number of seconds that elapse between the time that the execution of the *to-do-list* stops because of failure (error or timeout) and the time that the *to-do-failure* (list) is executed.

   1) Data type/values: Integer.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

j) **failure-execution-mode**. How the sequence of proxy operations specified in the *to-do-failure* attribute will be executed.

   1) Data type/values:

      i) ‘in sequence’. The sequence of proxy operations specified in the *to-do-failure* attribute will be executed in the sequence in which they appear.

      ii) ‘best effort’. No sequence is prescribed for the execution.

   3) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

k) **failure-limitation.** The limitations applied to triggering eventHandler managed objects; it identifies the conditions that force the eventHandler into the *disabled* state relative to the execution of the *to-do-failure* list.

   1) Data type/values:

      i) ‘once’. The *to-do-failure* list is to be executed only once. The eventHandler will be disabled for subsequent occurrences of the event.

      ii) ‘forever’. The *to-do-failure* list is to be executed for all occurrences of the event until the eventHandler is disabled.

      iii) ‘time window’. The *to-do-failure* list is to be executed for all occurrences of the event during the time window bounded by *to-do-failure-start* and *to-do-failure-stop*.

      iv) ‘max trigger’. The sequence of to-do-items specified in the *to-do-failure* attribute is to be executed for all occurrences of the event up to a maximum number of executions.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

l) **to-do-failure-start.** The start time of the time window for executing the *to-do-failure* list.

1) Data type/values: CCSDS time code.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

3) Dependencies: This attribute has meaning only when the value of *failure-limitation* equals 'time window'.

m) **to-do-failure-stop.** The stop time of the time window for executing the *to-do-failure* list.

1) Data type/values: CCSDS time code.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

3) Dependencies: This attribute has meaning only when the value of *failure-limitation* equals 'time window'.

n) **to-do-list.** A sequence of proxySet (5.12) and/or a proxyAction (5.11) managed objects. Each referenced proxy operation managed object identifies a target managed object and defines a proxy operation (SET or ACTION, respectively) to be invoked on that target object. The structure of this attribute and the behavior for the execution of a *to-do-list* are defined in 5.6.5.2.

1) Data type/values: Sequence of DNs.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

3) This attribute is conditional; if it is absent, the *to-do-list* shall be considered empty.

o) **to-do-failure.** A sequence proxySet (5.12) and/or a proxyAction (5.11) managed objects. Each referenced proxy operation managed object identifies a target managed object and defines a proxy operation (SET or ACTION, respectively) to be invoked on that target object. The resultant sequence of proxy operations are to be invoked only when the execution of the *to-do-list* stops with a failure (error or timeout). The behavior for the execution of *to-do-failure* is defined in 5.6.5.3.

1) Data type/values: Sequence of DNs.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

3) This attribute is conditional; if it is absent, there shall be no failure handling.

p) **enable-switch.** Enable or disable the eventHandler recognition of the occurrence of the specified event.

1) Data type/values:

i) ‘on’. Handling for the specified event is enabled.

ii) ‘off’. The event will be ignored.

2) Modifications: SET shall be enabled in both the definition and utilization phases.

q) **events-occurred**. A set of *event-identifiers* for events that occurred.

1) Data type/values: Integer. Initial value shall be ‘empty’.

2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

r) **to-do-list-executions**. The accumulated number of executions of the *to-do-list* until reporting time.

1) Data type/values: Integer. Initial value shall be ‘0’.

2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

s) **to-do-failure-executions**. The accumulated number of executions of *to-do-failure* until reporting time.

1) Data type/values: Integer. Initial value shall be ‘0’.

2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

t) **to-do-list-mean-execution-time**. The mean execution time (in msec) of *to-do-list.*

1) Data type/values: Integer. Initial value shall be ‘0’.

2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

u) **to-do-failure-mean-execution-time**. The mean execution time (in msec) of *to-do-failure.*

1) Data type/values: Integer. Initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

v) **enabled-time.** The accumulated time (in msec) the managed object is in the state *enabled* until reporting time.

1) Data type/values: Integer. Initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

w) **enabled-disabled-periods.** A set of enabled and disabled periods.

1) Data type/values: Initial value shall be 'empty'. Subsequent values shall be pairs of CCSDS time codes.

2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

x) **done-time.** The time when the managed object entered the state *done.*

1) Data type/values: CCSDS time code. Initial value shall be 'null'.

2) Modifications: SLE Complex Management may modify this attribute when the managed object is in the state *done*.

Refer to table 5-5 for the eventHandler managed object attribute value summary.

Table 5-5: Attribute Value Summary—EventHandler Managed Object

Attribute	Inherited From	Initial Value	Specified in
event-handler-mo-id	—	CM	CREATE.inv
event-handler-state	—	= waiting	CREATE.inv
event-identifier	—	UM	CREATE.inv
initiating-trigger-delay	—	UM	CREATE.inv
to-do-execution-mode	—	UM	CREATE.inv
to-do-limitation	—	UM	CREATE.inv
to-do-start	—	UM	CREATE.inv
to-do-stop	—	UM	CREATE.inv
failure-trigger-delay	—	UM	CREATE.inv
failure-execution-mode	—	UM	CREATE.inv
failure-limitation	—	UM	CREATE.inv
to-do-failure-start	—	UM	CREATE.inv
to-do-failure-stop	—	UM	CREATE.inv
to-do-list	—	UM	CREATE.inv
to-do-failure	—	UM	CREATE.inv
enable-switch	—	UM	CREATE.inv
events-occurred	—	= ‘empty’	as needed
to-do-list-executions	—	= 0	as needed
to-do-failure-executions	—	= 0	as needed
to-do-list-mean-execution-time	—	= 0	as needed
to-do-failure-mean-execution-time	—	= 0	as needed
enabled-time	—	= 0	as needed
enabled-disabled-periods	—	= ‘empty’	as needed
done-time	—	CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	UM	CREATE.inv

5.6.5.2 Behavior for the Execution of the *To-Do-List*

a) A *to-do-list* is executed as follows:

1) The proxy operations specified in the proxySet and/or proxyAction managed objects referenced by the *to-do-list* shall be invoked only after the specified event has occurred and the *initiating-trigger-delay* has elapsed.

2) If *to-do-execution-mode* is ‘in sequence’, the proxy operations shall be invoked one after the other in the sequence in which they appear in the *to-do-list*. The subsequent proxy operation shall be invoked only if the previous proxy operation succeeds (i.e., neither an error nor a timeout occurred). The execution of the *to-do-list* in ‘in sequence’ mode shall stop if a proxy operation fails.

3) If *to-do-execution-mode* is 'best effort', no sequence for invoking proxy operations shall be prescribed; however, all proxy operations shall be executed. The execution of the *to-do-list* shall stop when all proxy operations have been invoked (regardless of success, error or time-out).

b) A proxy SET operation is invoked as follows:

   1) A SET operation invocation shall be constructed according to the information specified in the proxySet managed object (5.12).

   2) The SET operation shall be invoked on the *target-mo*.

c) An proxy ACTION operation is invoked as follows:

   1) An ACTION operation invocation shall be constructed according to the information specified in the proxyAction managed object (5.11).

   2) The ACTION operation shall be invoked on the *target-mo*.

   3) The eventHandler managed object waits for the confirmation under timeout supervision.

   4) If the invocation is confirmed in time, the corresponding *proxy-operation-return* notification shall be appended to the *event-notification*. If the confirmation times out, the corresponding 'time-out-msg' shall be appended to the *event-notification*.

5.6.5.3 Behavior for the Execution of the *To-Do-Failure*

A *to-do-failure* is executed as follows:

a) *To-do-failure* shall be executed only when the execution of the *to-do-list* stops with a failure.

b) *To-do-failure* shall be executed in the same manner as the *to-do-list*.

c) If the execution of *to-do-failure* fails itself, the *to-do-failure-failed* shall be set to 'failed' and appended to the *event-notification*.

5.6.6 ACTIONS—EventHandler MO CLASS

a) An **enforce-to-do-list** action causes the unconditional execution of the *to-do-list.* If no *to-do-list* is present, the action has no effect.

b) No specific *action-parameters* are associated with this action.

5.6.7 STATE-RELATED BEHAVIOR—EventHandler MO CLASS

5.6.7.1 General

See figure 5-4 for the eventHandler state transition diagram. Since eventHandler managed objects are contained by servicePackage managed objects, the first two states of the definition phase, *waiting* and *under validation*, are the same as for the ServicePackage managed object class. Refer to 5.17.4 and 5.17.7 for a discussion of definition phase states and state transitions associated with the servicePackage managed object and its contained objects. We begin our discussion with the *validated* state.

NOTES

1 GET operation requests are accepted in all states.

2 No general rules are defined in this subsection for the acceptance or rejection of SET operation requests. They are state- and attribute-dependent and, hence, are part of the attributes description.

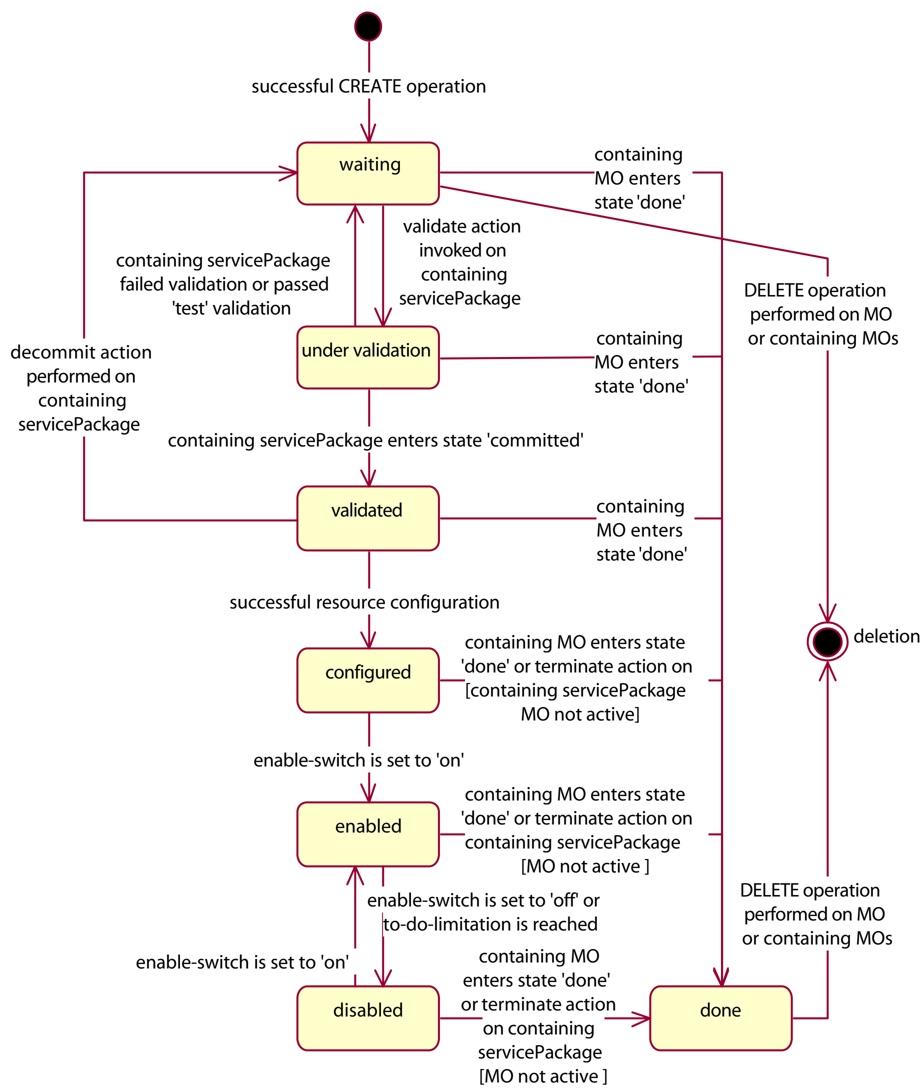


Figure 5-4: EventHandler State Transition Diagram

5.6.7.2 VALIDATED State

5.6.7.2.1 Conditions for the eventHandler managed object entering the state *validated*:

Successful validation of the containing servicePackage managed object, i.e., when containing servicePackage managed object enters the *committed* state.

5.6.7.2.2 Actions and operations accepted and rejected by the eventHandler managed object in the state *validated*

a) Rejects DELETE invocations.

b) Rejects a CREATE invocation for contained managed objects.

c) Rejects *enforce-to-do-list* action invocations.

5.6.7.2.3 Actions and operations on containing managed objects that affect the eventHandler managed object when in the state *validated*

a) *Decommit* action performed on the containing servicePackage managed object causes a state transition to *waiting*.

b) *Abort* action performed on the containing servicePackage managed object causes a state transition to *done*.

5.6.7.3 CONFIGURED State

5.6.7.3.1 Conditions for the eventHandler managed object entering the state *committed*

If the SLE Complex Management has established an association between the managed object and the resource, and the resource has been successfully configured according to the managed object's attribute values.

5.6.7.3.2 Actions and operations accepted and rejected by the eventHandler managed object in the state *configured*

a) Rejects DELETE invocations.

b) Rejects a CREATE invocation for contained managed objects.

c) Accepts an *enforce-to-do-list* action invocation, unless while executing the *to-do-list* (due to a previous *enforce-to-do-list* action).

5.6.7.3.3 Actions and operations on containing managed objects that affect the eventHandler managed object when in the state *committed*

a) *Terminate* action performed on the containing servicePackage managed object can have different consequences according to the current status of the eventHandler managed object:

1) If the eventHandler managed object is not executing the *to-do-list*, the *terminate* action causes a state transition to *done*.

2) If the eventHandler managed object is executing the *to-do-list* after an *enforce-to-do-list* action invocation, then the *terminate* action has no immediate effect on the eventHandler managed object. It will transition to the state *done* when the containing servicePackage enters the state *done*.

b) *Abort* action invoked on the containing servicePackage managed object causes a state transition to *done*.

5.6.7.4 ENABLED State

5.6.7.4.1 General

When in the state *enabled*, the eventHandler managed object:

a) checks whether the specified event has occurred;

b) executes the to-do-list if the specified event has occurred and issues the respective *event-notifications*;

c) ignores further occurrences of the specified event while executing the *to-do-list*.

5.6.7.4.2 Conditions for the eventHandler managed object entering the state *enabled*

If *enable-switch* is set to 'on' when the eventHandler managed object is either in the state *configured* or the state *disabled*.

5.6.7.4.3 Actions and operations accepted and rejected by the eventHandler managed object in the state *enabled*

a) Rejects DELETE invocations.

b) Rejects a CREATE invocation for contained managed objects.

c) Accepts an *enforce-to-do-list* action invocation, unless while executing the *to-do-list*.

5.6.7.4.4 Actions and operations on containing managed objects that affect the eventHandler managed object when in the state *enabled*

a) *Terminate* action performed on the containing servicePackage managed object can have different consequences according to the current status of the eventHandler managed object:

1) If the eventHandler managed object is not executing the *to-do-list*, the *terminate* action causes a state transition to *done*.

2) If the eventHandler managed object is executing the *to-do-list* (either after the specified event occurrence or after an *enforce-to-do-list* action invocation), then

the *terminate* action has no immediate effect on the eventHandler managed object. It will transition to the state *done* when the containing servicePackage enters the state *done*.

b) *Abort* action performed on the containing servicePackage managed object causes a state transition to *done*.

5.6.7.5 DISABLED State

5.6.7.5.1 General

When in the state *disabled*, the eventHandler managed object ignores the specified event.

5.6.7.5.2 Conditions for the eventHandler managed object entering the state *disabled*

a) If *enable-switch* is set to 'off' when the eventHandler managed object is in the state *enabled*.

b) If the *to-do-limitation* specifies 'max-trigger', and the *to-do-list* has already been executed the maximum number of times specified.

c) If the *to-do-limitation* specifies a 'time-window' and this window is expired.

5.6.7.5.3 Actions and operations accepted and rejected by the eventHandler managed object in the state *disabled*

a) Rejects DELETE invocations.

b) Rejects a CREATE invocation for contained managed objects.

c) Accepts an *enforce-to-do-list* action invocation, unless while executing the *to-do-list* (due to a previous *enforce-to-do-list* action).

5.6.7.5.4 Actions and operations on containing managed objects that affect the eventHandler managed object when in the state *enabled*

a) *Terminate* action performed on the containing servicePackage managed object can have different consequences according to the current status of the eventHandler managed object:

   1) If the eventHandler managed object is <u>not</u> executing the *to-do-list*, the *terminate* action causes a state transition to *done*.

   2) If the eventHandler managed object is executing the *to-do-list* after an *enforce-to-do-list* action invocation, then the *terminate* action has no immediate effect on the eventHandler managed object. It will transition to the state *done* when the containing servicePackage enters the state *done*.

b) *Abort* action performed on the containing servicePackage managed object causes a state transition to *done*.

5.6.7.6 DONE State

5.6.7.6.1 Conditions for the eventHandler managed object entering the state *done*

a) When the containing servicePackage managed object enters the state *done.*

b) When a *terminate* action is performed on the containing servicePackage managed object and the eventHandler managed object is in state *configured*, *enabled* or *disabled* and not executing the *to-do-list* or the *to-do-failure*.

5.6.7.6.2 Actions and operations accepted and rejected by the eventHandler managed object in the state *done*

a) Accepts DELETE invocations.

b) Rejects a CREATE invocation for contained managed objects.

c) Rejects *enforce-to-do-list* action invocations.

5.6.7.6.3 Actions and operations on containing managed objects that affect the eventHandler managed object when in the state *done*

DELETE operation performed on the containing servicePackage managed object causes the eventHandler managed object to be deleted.

5.6.8 NOTIFICATIONS—EventHandler MO CLASS

The eventHandler managed object issues the following unique notification:

a) **Event-notification**. The event for which notification is issued. An *event-notification* may contain the following additional information:

   1) **proxy-operation-return-set**. A sequence of *proxy-operation-return* data structures. If no *to-do-list* is present or all managed objects timed out, the set is empty. If a *to-do-list* is present, the set contains zero, one, or more *proxy-operation-returns*. The sequence in which the *proxy-operation-returns* appear in the *event-notification* has no significance. A *proxy-operation-return* has the following elements:

      i) **Target-mo.** The DN of the managed object that sent the return.

      ii) **Invoke-id.** The identification of the invocation of the respective proxy operation.

      iii) **Return-msg.** The actual return as it is sent by the *target-mo.* It may represent either a result or an error.

   2) **time-out-msg-set**. A sequence of *time-out-msg data* structures. If no target managed object timed out, the set is empty. The sequence in which the *time-out-*

msg appear in the *event-notification* has no significance. A *time-out-msg* has the following elements:

i) **Target-mo.** The DN of the managed object that timed out.

ii) **Invoke-id.** The identification of the invocation of the respective proxy operation.

3) **to-do-failure-failed.** Failure status of *target-mo*. This item is set to 'failed' if any *target-mo* in the *to-do-failure* fails, either by error or timeout.

5.6.9 VALIDATION ASPECTS—EventHandler MO CLASS

a) The managed objects specified as targets of *to-do-list* or *to-do-failure* must exist. If this constraint is not met at validation time, a 'completeness defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

b) The proxy operations specified in *to-do-list* or *to-do-failure* must match the attributes or actions defined for their respective targets. If this constraint is not met at validation time, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

5.7 F-proto-vcdu-prod MANAGED OBJECT CLASS

NOTE – Although the *Cross Support Reference Model* (reference [1]) specifies a set of forward AOS services and functional groups, the current suite of SLE services do not contain any forward AOS services. This managed object class definition subsection is a placeholder for one that will be needed when forward AOS services are specified.

5.7.1 PURPOSE

This managed object embodies the management interface to the functionality that produces a forward protoVCDU channel (reference [5]). This production is executed as part of the Forward AOS Virtual Channel (VC) Data Insertion functional group.

5.7.2 INHERITANCE—F-proto-vcdu-prod MO CLASS

This managed object class is derived from and inherits the properties of the Production managed object class.

5.8 F-transferService MANAGED OBJECT CLASS

5.8.1 PURPOSE

This managed object represents the resources needed to either provide or use forward transfer services. See 5.20 for a general discussion of transfer service provision and use.

5.8.2 INHERITANCE—F-transferService MO CLASS

This managed object class is derived from and inherits the properties of the TransferService managed object class.

5.8.3 OBJECTS CONTAINED—F-transferService MO CLASS

This managed object class is an abstract class; thus it cannot be instantiated and containment relationships do not apply.

5.8.4 STATES—F-transferService MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the TransferService managed object class.

5.8.5 ATTRIBUTES—F-transferService MO CLASS

This managed object class has the following specific attribute: **f-delivery-mode.** The *delivery-mode* is defined in references [1] and [E9]).

a) Data type/values:

   1) 'forward online';

   2) 'forward offline'.

b) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

Refer to table 5-6 for the F-transferService managed object attribute value summary.

Table 5-6: Attribute Value Summary—F-transferService Managed Object

Attribute	Inherited From	Initial Value	Specified in
f-delivery-mode	—	by UM	CREATE.inv
current-ts-version	—	by UM	CREATE.inv
lower-bound-reporting-period	—	by UM	CREATE.inv
time-out-period	—	by UM	CREATE.inv
transfer-service-state	—	= *waiting*	—
association-initiation-responsibility	—	by UM	CREATE.inv
responder-port-identifier	—	by UM	CREATE.inv
local-sle-entity-identifier	—	by CM	—
peer-sle-entity-access-list	—	by UM	CREATE.inv
scheduled-period-start-time	—	by UM	CREATE.inv
scheduled-period-stop-time	—	by UM	CREATE.inv
first-data-unit-time	—	by CM	when first data unit is transferred
last-data-unit-time	—	by CM	when last data unit is transferred
service-instance-start	—	= 'null'	when first service instance starts
service-instance-stop	—	= 'null'	when last service instance stops
provision-time	—	= 0	as needed
failed-binds	—	= 0	as needed
unavailable-time	—	= 0	as needed
done-time	—	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

5.8.6 ACTIONS—F-transferService MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the TransferService managed object class.

5.8.7 STATE-RELATED BEHAVIOR—F-transferService MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the TransferService managed object class.

5.8.8 NOTIFICATIONS—F-transferService MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the TransferService managed object class.

5.8.9 VALIDATION ASPECTS—F-transferService MO CLASS

No specific validation aspects are defined for this managed object class. However, validation aspects are inherited from the TransferService managed object class.

5.9 NotificationLog MANAGED OBJECT CLASS

5.9.1 PURPOSE

5.9.1.1 General

a) Notifications issued by the managed objects in a servicePackage (or its contained managed objects) shall be recorded if:

   1) no telecommunication association exists between SLE Utilization Management and SLE Complex Management;

   2) telecommunication is available, but the performer SLE Utilization Management has returned neither a result nor an error;

   3) the NOTIFY operation (see 3.4.2.5) fails.

   NOTE 1 – This records the attempt to deliver the notification.

b) A notificationLog managed object holds statistics about the recorded notifications.

c) Using the *dump* action, SLE Utilization Management may retrieve logged notifications when a telecommunication association with SLE Complex Management is re-established.

NOTE 2 – Notifications can be retrieved in a staggered manner, thus avoiding telecommunication blockage by transfer of a potentially large number of notifications. Logged notifications are accessible only via the *dump* action.

d) A tag identifies the reason the notification was logged.

e) When retrieved, the actual notification appears in the *dump* action return exactly as if it had been sent directly via a NOTIFY invocation.

NOTE 3 – The support of retrieval filters (e.g., managed object class, time, severity, etc.) and of their possible logical combinations is for further study.

5.9.1.2 Perception by SLE Utilization Management

In order to prevent a loss of logged notifications, SLE Utilization Management shall monitor the value of the *fill-now* attribute so that notifications of interest may be retrieved before they are overwritten.

5.9.1.3 Perception by SLE Complex Management

a) SLE Complex Management shall provide a suitable recording resource for storing the agreed number of notifications (the *max-notification-log-size* in the serviceAgreement) and that supports the behavior of the NotificationLog managed object class.

b) If a new notification is to be logged but the agreed number of notifications has already been reached, the oldest notification shall be overwritten and the *notifications-lost* attribute incremented by 1. The overwritten notification is lost and cannot be recovered.

NOTE – This draft Recommendation does not prescribe the format in which notifications are stored inside the Complex. However, when a notification is delivered to SLE Utilization Management via a *dump* action, the format and contents of *notification-message* are identical to the format and contents of *notification-message* in a NOTIFY operation.

5.9.1.4 Deletion Aspects

A notificationLog managed object is deleted only if the servicePackage managed object that contains it is deleted.

5.9.2 INHERITANCE—NotificationLog MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

5.9.3 OBJECTS CONTAINED—NotificationLog MO CLASS

An instance of this concrete managed object class contains no other managed objects.

5.9.4 STATES—NotificationLog MO CLASS

A notificationLog managed object has no states.

5.9.5 ATTRIBUTES—NotificationLog MO CLASS

This managed object class has the following specific attributes:

a) **notification-log-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (see annex C).

   1) Data type/values: Character string.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **total-notification-records**. The total number of notifications logged and available for retrieval.

1) Data type/values: Integer. Initial value shall be ‘0’.

2) Modifications: SLE Complex Management may modify the value during the utilization phase when notifications are logged or dumped. SET shall be disabled in both the definition and utilization phases.

c) **oldest-notification.** The issue time of the oldest *notification-message* logged.

1) Data type/values: CCSDS time code.

2) Modifications: SLE Complex Management may modify the value during the utilization phase when notifications are logged, dumped or cleared. SET shall be disabled in both the definition and utilization phases.

d) **last-notification**. The generation time of the most recent *notification-message* logged.

1) Data type/values: CCSDS time code.

2) Modifications: SLE Complex Management may modify the value during the utilization phase when notifications are logged, dumped or cleared, as follows:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

e) **fill-now.** The percentage up to which the storage capacity for notification records is used.

1) Data type/values: Integer value closest to the computed (real) value. Initial value shall be ‘0’.

2) Modifications: SLE Complex Management may modify the value during the utilization phase when notifications are logged, dumped or cleared, as follows:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

f) **notifications-lost**. The number of notification records that were overwritten because the maximum number of notifications was exceeded.

1) Data type/values: Integer. Initial value shall be ‘0’.

2) Modifications: SLE Complex Management may increment the value during the utilization phase when notifications are logged. The value is reset to ‘0’ after a *dump* or *clear* action has been performed.

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

Refer to table 5-7 for the notificationLog managed object attribute value summary.

Table 5-7: Attribute Value Summary—NotificationLog Managed Object

Attribute	Inherited From	Initial Value	Specified In
notification-log-mo-id	—	by CM	CREATE.inv
total-notification-records	—	= 0	No
oldest-notification	—	by CM	No
last-notification	—	by CM	No
fill-now	—	by CM	No
notifications-lost	—	= 0	No
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

5.9.6 ACTIONS—NotificationLog MO CLASS

5.9.6.1 Clear Action

a) A **clear** action is used to remove all logged records from the storage resource. After a *clear* action, the notificationLog managed object's attributes must be updated accordingly.

b) No specific *action-parameters* are associated with this action.

5.9.6.2 Dump Action

a) A **dump** action is used to retrieve a portion of the logged notification records and transfer them to SLE Utilization Management.

   1) The dumped notification records shall be sorted according to generation time and the value of the *dump-direction action-parameter*.

   2) Dumped notification records shall be removed from the notification log and the notificationLog managed object shall be updated accordingly.

   3) While a *dump* action is being performed (from the invocation through return), no further *dump* action can be invoked.

b) The *dump* action invocation shall have the following *action-parameters*:

   1) **portion-size**. The portion of notifications to be dumped.

      i) If 'all' is specified, all notifications shall be dumped.

      ii) If a positive integer is specified, a portion of notifications shall be dumped. A portion shall be limited to the specified number of notifications.

   2) **dump-direction**. The order in which to dump notifications.

i) If Last-In-First-Out (LIFO) is specified, the notification record with the most recent *notification-message* shall become first in the sequence delivered.

ii) If First-In-First-Out (FIFO) is specified, the notification record with the oldest *notification-message* shall become first in the sequence delivered.

c) A *dump* action result carries the information item **notifications-retrieved**. This item shall consist of a sequence of **notification records**. The sequence shall be ordered according to the *dump-direction*. The sequence contains at most the number of notification records requested in *portion-size.* If no notifications are available for retrieval, the sequence is empty. Each *notification record* shall consist of the following:

1) **log-reason**. The reason the notification record was logged. Possible values are as follows:

i) 'no link with UM';

ii) 'no return by UM';

iii) 'NOTIFY error'.

2) **notification-message.** The notification message, formatted exactly as defined for the respective NOTIFY invocation.

5.9.7 STATE-RELATED BEHAVIOR—NotificationLog MO CLASS

This managed object class has no state-related behavior.

5.9.8 NOTIFICATIONS—NotificationLog MO CLASS

A notificationLog managed object issues no notifications.

5.9.9 VALIDATION ASPECTS—NotificationLog MO CLASS

a) If at its validation time, a servicePackage does not contain a notificationLog managed object, SLE Complex Management assumes that notification logging is not required.

b) If, in the serviceAgreement managed object, *max-notification-log-size* is set to '0' and a notificationLog managed object is present, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records.*

5.10 Production MANAGED OBJECT CLASS

5.10.1 PURPOSE

5.10.1.1 General

According to the *Cross Support Reference Model* (reference [1]), a Functional Group is an abstraction of the Complex resources that process SLE data channels as required to deliver

SLE transfer services. SLE transfer service production is the performance of transformations between the RF carrier channel and the SLE data channels accessible by SLE transfer service users. SLE transfer service production is accomplished by a sequence of production steps, each of which provides a value-added transformation to the SLE data channel that is input to it. These production steps are performed by the various functional groups defined in the reference [1]. In general, a given production step takes a specific SLE data channel as input and covers its processing for multiple instances of a variety of SLE transfer services.

a) In order to quickly adapt an active production step to changing requirements of the MDOS, SLE Complex Management shall offer online control to SLE Utilization Management. A production resource shall be associated with a corresponding production managed object that embodies the possible controls.

NOTE 1 – To understand the concept of production managed objects, it is absolutely essential to clearly distinguish between a <u>production resource</u> and a <u>production managed object</u>. For any SLE-PDU processing, a Complex must allocate a suitable <u>production resource</u>. A <u>production managed object</u> is associated with this resource only if external control or monitoring of the processing resource is required.

b) Control of a production managed object affects the behavior of its production resource; however, side-effects may extend beyond and ripple through to subsequent processing of SLE data channels performed by other functional groups or by Mission End Users.

NOTE 2 – All managed objects contained in a given managed object form a ‘processing subtree’ relative to the managed object that contains them.

c) In order to guarantee that a control effect is limited to a single ‘processing subtree’ of a given production managed object (of a derived class), the Complex must provide a number of independently operating production resources of the same type, one at the root of each processing subtree. This requirement can be expressed by the fact that:

   1) All managed objects in a ‘processing subtree’ are contained by a single production managed object. This managed object, in turn, is associated with one of the independent functional group resources.

   2) Multiple production managed objects of the same type can co-exist and be contained by a single managed object.

d) The common properties of all controlling/monitoring managed objects are summarized in the abstract Production managed object class.

NOTE 3 – From this abstract class, specific classes are derived for each type of production step performed in the various functional groups (reference [1]). The derived classes are listed and their corresponding functional groups are shown in table 5-8.

Table 5-8: Production Derived Classes

Derived class	Functional Group Type
R-cf-prod	Return Frame Processing
R-fsh-prod	Return Frame Processing
R-ocf-prod	Return Frame Processing
R-sp-prod	Return Frame Data Extraction
R-bitstream-prod*	Return Frame Data Extraction
F-pca-pdu-prod*	Forward AOS Space Link Processing
F-c/vcdu-prod*	Forward AOS Space Link Processing
F-proto-vcdu-b_pdu-prod*	Forward AOS VC Data Insertion
F-proto-vcdu-m_pdu-prod*	Forward AOS VC Data Insertion
F-proto-vcdu-vca-prod*	Forward AOS VC Data Insertion
F-vc-multiplex-prod	Forward TC VC Data Insertion
F-tc-vc-prod	Forward TC VC Data Insertion
F-vc-seg-prod	Forward TC VC Data Insertion
F-tc-session-prod	Forward TC Space Link Processing
F-cltu-prod	Forward CLTU Generation
R-af-prod	Return Space Link Processing
R-insert-prod*	Return Space Link Processing
R-frameDataField-prod	Return Frame Data Extraction
* This production managed object represents resources that are needed to support a service of the *Cross Support Reference Model* (reference [1]), but which are not contained in the Version 1 suite of SLE services.	

e) Setting one or more of the associated production managed object's attributes controls the production resource. Getting one or more of the production managed object's attributes enables the user to monitor the production resource.

5.10.1.2 Perception by SLE Utilization Management

a) If the character of a mission requires quick adaptation of data channel processing, SLE Utilization Management creates an appropriate production managed object.

b) The default parameterization of functional group resources can be corrected before their activation by setting the appropriate attributes of a production managed object.

c) Setting one or more production managed object attributes can influence ongoing data channel processing.

d) The containment structure between managed objects must be exploited such that the desired processing subtrees are configured, so that the effects of a control only propagate to the Ground System components represented by contained managed objects.

5.10.1.3 Perception by SLE Complex Management

a) SLE Complex Management shall be responsible for:

   1) associating the production managed object with a suitable production resource;

   2) ensuring that each attribute of a production managed object is associated with a corresponding controllable and/or observable element of the production resource;

   3) ensuring that the attributes of a derived class production managed object are initially set up according to the rules specified in the respective derived class description.

b) If the servicePackage is *validated*, SLE Complex Management shall:

   1) enable and disable the setting of the attributes supported by the Complex according to the life cycle of their associated local resources;

   2) permanently disable the setting of attributes relating to resources that will never be activated for processing SLE data channels (i.e., resources that are not required for the specific services to be provided).

5.10.1.4 Deletion Aspects

Subsection 5.10.7 contains information about the states in which a production managed object can be deleted.

5.10.2 INHERITANCE—Production MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

5.10.3 OBJECTS CONTAINED—Production MO CLASS

NOTE 1 – This managed object class is an abstract class; thus it cannot be instantiated and containment relationships do not apply.

An instance of any of the production derived classes shall contain the following:

a) Zero, one, or more ‘service provided’ managed objects. The presence of a contained ‘service provided’ managed object indicates that any control that may be exerted via the production managed object may have an immediate impact on the behavior of the corresponding ‘service provided’ resource. The possible controls and their impact are specified in the CCSDS Recommendation for the related SLE transfer service.

b) Zero, one, or more channelStored managed objects. The presence of a contained channelStored managed object indicates that any control which may be exerted via the production managed object may have an immediate impact on the behavior of the corresponding Channel Stored resource. The possible controls and their impact are specified in the CCSDS Recommendation for the related SLE transfer service.

c) Zero, one, or more other production managed objects. The presence of a contained production managed object indicates that the Complex has to perform multi-stage processing of SLE data channels to produce the desired *data-product*. Any control that may be exerted via the production managed object may have an immediate impact on the behavior of the 'adjacent' production resources.

NOTE 2 – Specific derived classes of the production managed object class may allow for containment of additional managed objects, including other production managed objects.

5.10.4 STATES—Production MO CLASS

a) In the definition phase, a production managed object shall exist in one of the following states:

1) **Waiting**. The production managed object is created or the servicePackage managed object that contains it has entered the state *waiting* (e.g., after a decommit action).

2) **Under validation**. The servicePackage managed object is undergoing validation. The production managed object and any contained managed objects are static until validation is complete.

3) **Validated**. The production managed object is confirmed by the successful validation of the servicePackage managed object that contains it. All pertinent Complex-internal resources are allocated to the managed object.

b) In the utilization phase, a production managed object shall exist in one of the following states:

1) **Configured**. An association between the managed object and the resource has been established and the resource has been successfully configured according to the managed object's attribute values. The production process is not yet capable of supplying or consuming SLE-SDUs.

NOTE – This state corresponds to the value 'configured' of the *production-status*.

2) **Operational**. The production process is now capable of supplying and consuming SLE-SDUs.

NOTE – This state corresponds to the value 'operational' of the *production-status*.

3) **Unavailable**. Service production is not possible for Complex-internal reasons.

NOTE – This state corresponds to the value 'interrupted' and 'halted' of the *production-status*.

4) **Completed**. The supply of SLE-SDUs to the associated production resources has ended. The production resources have been released.

5) **Done**. All contained managed objects have transitioned to *done*.

5.10.5 ATTRIBUTES—Production MO CLASS

This managed object class has the following specific attributes:

a) **production-state**. The state of the production resource associated with the production managed object. Only SLE Complex Management can set this attribute.

   1) Data type/values: See 5.10.4.

   2) Modifications: SLE Complex Management may modify the value at any time. SET shall be disabled in both the definition and utilization phases.

b) **data-units-consumed-for-processing**. The number of SLE channel data units consumed as input to production processing, per SLE data channel.

   1) Data type/values: Integer. Initial value shall be ‘0’.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

c) **processed-successfully**. The number of SLE channel data units successfully processed, per SLE data channel.

   1) Data type/values: Integer. Initial value shall be ‘0’.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

d) **data-units-produced**. The number of SLE channel data units produced as output of production processing.

   1) Data type/values: Integer. Initial value shall be ‘0’.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

e) **number-invocations**. The accumulated number of operation invocations until reporting time.

1) Data type/values: Integer. Initial value shall be ‘0’.

2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

f) **unavailable-time.** The accumulated time (in seconds) the managed object was in the state *unavailable* until reporting time.

1) Data type/values: Integer. Initial value shall be ‘0’.

2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

g) **operational-time.** The accumulated time (in seconds) the managed object was in the state *operational* until reporting time.

1) Data type/values: Integer.

2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

h) **production-start-time**. The start time.

1) Data type/values: CCSDS time code.

2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

i) **production-stop-time**. The stop time.

1) Data type/values: CCSDS time code.

2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

j) **done-time**. The time when the managed object entered the state *done*.

   1) Data type/values: CCSDS time code.

   2) Modifications: SLE Complex Management may modify this attribute using a SET invocation during the utilization phase.

NOTE – Each derived class of the Production class features additional attributes specific to that class.

Refer to table 5-9 for the production managed object attribute value summary.

Table 5-9: Attribute Value Summary—Production Managed Object

Attribute	Inherited From	Initial Value	Specified In
production-state	—	= waiting	—
data-units-consumed-for-processing	—	= 0	as needed
processed-successfully	—	= 0	as needed
data-units-produced	—	= 0	as needed
number-invocations	—	= 0	as needed
unavailable-time	—	= 0	as needed
operational-time	—	= 0	as needed
production-start-time	—	by CM	when production starts
production-stop-time	—	by CM	when production stops
done-time	—	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

5.10.6 ACTIONS—Production MO CLASS

An **abort** action inhibits production even after successful validation. No specific *action-parameters* are associated with this action.

5.10.7 STATE-RELATED BEHAVIOR—Production MO CLASS

5.10.7.1 General

See figure 5-5 for the production state transition diagram. Since production managed objects are ultimately contained by servicePackage managed objects, the first two states of the definition phase, *waiting* and *under validation*, are the same as for the ServicePackage managed object class. Refer to 5.17.4 and 5.17.7 for a discussion of definition phase states and state transitions associated with the servicePackage managed object and its contained objects. We begin our discussion with the *validated* state.

NOTES

1 GET operation requests are accepted in all states.

2 No general rules are defined in this subsection for the acceptance or rejection of SET operation requests. They are state- and attribute-dependent and, hence, are part of the attributes description.

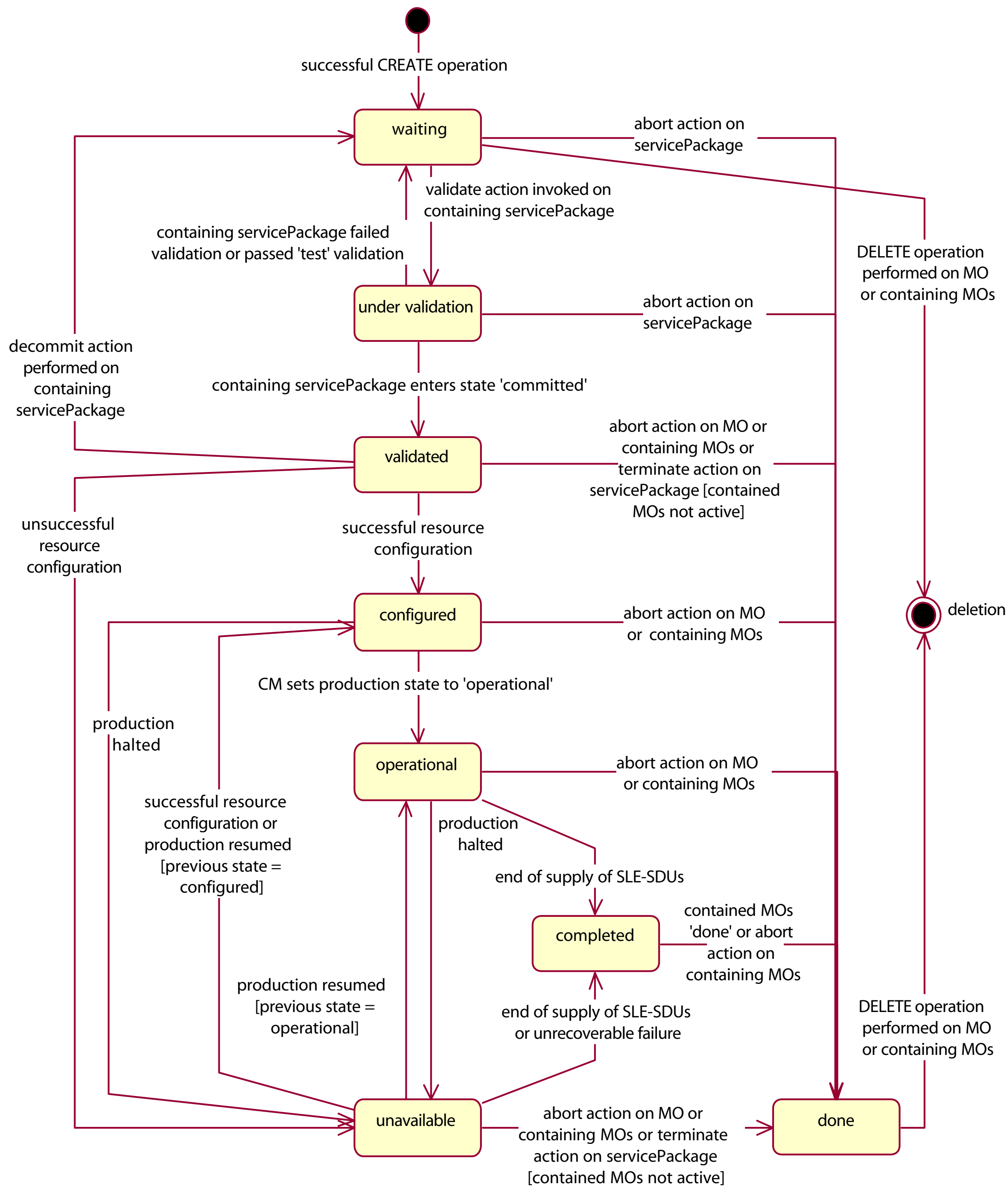


Figure 5-5: Production State Transition Diagram

5.10.7.2 VALIDATED State

5.10.7.2.1 General

When the state *validated* is entered, the specific attributes for an instance of a derived class are set to the initial values applicable to the associated functional group resource.

5.10.7.2.2 Conditions for the production managed object entering the state *validated*:

If the containing servicePackage managed object enters the *committed* state.

5.10.7.2.3 Actions and operations accepted and rejected by the production managed object in the state *validated*

a) Rejects DELETE invocations.

b) Rejects a CREATE invocation for contained managed objects.

c) Accepts an *abort* action invocation.

5.10.7.2.4 Actions and operations on containing managed objects that affect the production managed object when in the state *validated*

a) *Decommit* action invoked on the containing servicePackage managed object causes the production managed object to transition to state *waiting*.

b) *Terminate* action invoked on the containing servicePackage managed object causes the production managed object to transition to state *done* if none of its contained managed objects is active (the states in which managed objects are considered 'active' are listed in 4.10).

c) *Abort* action invoked on the containing managed objects causes the production managed object to transition to state *done.*

d) *Terminate* action performed on the containing channelStored managed object causes the production managed object to transition to state *done*.

5.10.7.3 CONFIGURED State

5.10.7.3.1 General

When the state *configured* is entered, the attribute values specified in the associated production managed object overwrite the initial setting of the associated production resource.

5.10.7.3.2 Conditions for the production managed object entering the state *configured*

a) If the production managed object is successfully associated with a production resource, and the production resource is successfully configured.

b) If production is resumed when production managed object is in state *unavailable,* and if the *production-state* before transitioning to *unavailable* was *configured*.

5.10.7.3.3 Actions and operations accepted and rejected by the production managed object in the state *configured*

a) Rejects DELETE invocations.

b) Rejects a CREATE invocation for contained managed objects.

c) Accepts an *abort* action invocation.

5.10.7.3.4 Actions and operations on containing managed objects that affect the production managed object when in the state *configured*

a) *Abort* action invoked on the containing managed objects causes a state transition to *done.*

b) *Terminate* action performed on the containing channelStored managed object causes the production managed object to transition to state *done*.

5.10.7.4 OPERATIONAL State

5.10.7.4.1 Conditions for the production managed object entering the state *operational*

a) When SLE Complex Management activates the production resource for SLE-PDU processing, the *production-state* is set to 'operational'.

NOTE – The conditions for setting the production state to *operational* are determined by SLE Complex Management in order to meet the start and stop times of the entities that supply to, and consume from, the production resources associated with the production managed object. The time at which SLE Complex Management makes the production managed object *operational* depends on the entities involved in providing the required service, hence, it is service-dependent. For each type of service it should be possible to specify two time limits: latest time the production managed object can transition to *operational,* and earliest time the production managed object can transition to *completed*.

b) If production is resumed when production managed object is in state *unavailable,* and if the *production-state* before transitioning to *unavailable* was *operational*.

5.10.7.4.2 Actions and operations accepted and rejected by the production managed object in the state *operational*

a) Rejects DELETE invocations.

b) Rejects a CREATE invocation for contained managed objects.

c) Accepts an *abort* action invocation.

5.10.7.4.3 Actions and operations on containing managed objects that affect the production managed object when in the state *operational*

a) *Abort* action performed on the containing managed objects causes a state transition to *done*.

b) *Terminate* action performed on the containing channelStored managed object causes the production managed object to transition to state *done*.

5.10.7.5 UNAVAILABLE State

5.10.7.5.1 General

When in the state *unavailable*, the production managed object waits for production to resume.

5.10.7.5.2 Conditions for the production managed object entering the state *unavailable*:

a) If production is halted, because:

   1) the associated production resource cannot be allocated, or becomes unable to process SLE data channels;

   2) the association between the production resource and its controlling production managed object is destroyed.

5.10.7.5.3 Actions and operations accepted and rejected by the production managed object in the state *unavailable*

a) Rejects a DELETE invocation.

b) Rejects a CREATE invocation for contained managed objects.

c) Accepts an *abort* action invocation.

5.10.7.5.4 Actions and operations on containing managed objects that affect the production managed object when in the state *unavailable*

a) *Abort* action invoked on the containing managed objects causes a state transition to *done.*

b) *Terminate* action performed on the containing servicePackage managed object causes the production managed object to transition to state *done* if none of its contained managed objects is active (the states in which managed objects are considered ‘active’ are listed in 4.10).

c) *Terminate* action performed on the containing channelStored managed object causes the production managed object to transition to state *done*.

5.10.7.6 COMPLETED State

5.10.7.6.1 General

When in the state *completed*, the production managed object is dissociated from its production resource.

5.10.7.6.2 Conditions for the production managed object entering the state completed

a) When the entities that supply the production resources associated with the production managed object are no longer active.

NOTE – The conditions for setting the production state to *completed* are determined by SLE Complex Management in order to meet the start and stop times of the entities that supply to, and consume from, the production resources associated with the production managed object. The time at which SLE Complex Management makes the production managed object *completed* depends on the entities involved in providing the required service, hence, it is service-dependent. For each type of service it should be possible to specify two time limits: latest time the production managed object can transition to *operational,* and earliest time the production managed object can transition to *completed*.

b) When no recovery from corruption can be achieved.

5.10.7.6.3 Actions and operations accepted and rejected by the production managed object in the state *completed*

a) Rejects DELETE invocations.

b) Rejects a CREATE invocation for contained managed objects.

c) Rejects an *abort* action invocation.

5.10.7.6.4 Actions and operations on containing managed objects that affect the production managed object when in the state *completed*

a) *Abort* action performed on the containing managed objects causes the production managed object to transition to state *done*.

b) *Terminate* action performed on the containing channelStored managed object causes the production managed object to transition to state *done*.

5.10.7.7 DONE State

5.10.7.7.1 General

When the state *done* is entered, the production managed object is dissociated from its production resource.

5.10.7.7.2 Conditions for the production managed object entering the state *done*

a) If all of contained MOs have entered state *done*, when managed object is in state *completed*.

b) If a *terminate* action is performed on the containing servicePackage managed object when the production managed object is in state *validated* or *unavailable,* and none of the contained managed objects is active (the states in which managed objects are considered 'active' are listed in 4.10).

c) When an *abort* action is performed on the managed object or on the containing managed objects.

5.10.7.7.3 Actions and operations accepted and rejected by the production managed object in the state *done*

a) Accepts DELETE invocations.

b) Rejects a CREATE invocation for contained managed objects.

c) Rejects an *abort* action invocation.

5.10.7.7.4 Actions and operations on containing managed objects that affect the production managed object when in the state *done*

DELETE operation performed on the containing managed objects causes the production managed object to be deleted.

5.10.8 NOTIFICATIONS—Production MO CLASS

A production managed object issues the following state transition notifications:

a) **production halted**. This indicates that the managed object has transitioned from either the state *configured* or *operational* to the state *unavailable*. This notification contains the additional information:

   Reason for interruption. An implementation-specific value within a specific Complex. The value is provided by SLE Complex Management. It is a character string.

b) **production resumed**. This indicates that the managed object has transitioned from the state *unavailable* to the state *configured* or *operational*.

c) **unrecoverable failure**. The managed object has transitioned from the state *unavailable* to *completed*.

NOTE – Derived classes may provide additional notifications.

5.10.9 VALIDATION ASPECTS—Production MO CLASS

a) If at validation time a mandatory production managed object is missing, a ‘completeness defect’ is added to the containing servicePackage managed object’s set of *defect-records*.

b) If at validation time an existing production managed object does not match the dataChannelProfile managed object, an ‘inconsistency defect’ shall be added to the containing servicePackage managed object’s set of *defect-records*.

c) At least one managed object shall be contained by a production managed object. If not, a ‘completeness defect’ shall be added to the containing servicePackage managed object’s set of *defect-records*.

5.11 ProxyAction MANAGED OBJECT CLASS

5.11.1 PURPOSE

A proxyAction managed object specifies a target managed object and an ACTION operation that is to be invoked on that target object when the triggering event or failure condition associated with the containing eventHandler managed object occurs. These triggering events and failure conditions are specified as part of the eventHandler managed object (5.6) that contains the proxyAction managed object.

5.11.2 INHERITANCE—ProxyAction MO CLASS

This managed object class is derived from and inherits the attributes of the SLEManagementEntity managed object class.

5.11.3 OBJECTS CONTAINED—ProxyAction CLASS

An instance of this concrete managed object class contains no other managed objects.

5.11.4 STATES—ProxyAction MO CLASS

The ProxyAction managed object has no states.

5.11.5 ATTRIBUTES—ProxyAction MO CLASS

This managed object class has the following specific attributes:

a) **proxy-action-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (see annex C).

   1) Data type/values: Character string.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **target-mo.** A DN designating the target managed object, i.e., the managed object affected by the ACTION operation.

   1) Data type/values: DN.

   2) Modifications: SET shall be enabled when the containing eventHandler managed object is in the state *waiting*.

c) **target-action-id.** The target action to be invoked of the managed object designated by target-mo.

   1) Data type/values: See 3.4.2.4.2(f).

   2) Modifications: SET shall be enabled when the containing eventHandler managed object is in the state *waiting*.

d) **action-parameter-pairs.** A set of *action-parameter-pairs*. An *action-parameter-pair* is an *action-parameter* identifier paired with its value.

   1) Data type/values: See 3.4.2.4.2(g).

   2) Modifications: SET shall be enabled when the containing eventHandler managed object is in the state *waiting*.

Refer to table 5-10 for the proxyAction managed object attribute summary.

Table 5-10: Attribute Summary—ProxyAction Managed Object

Attribute	Inherited From	Initial Value	Specified In
proxy-action-mo-id	—	by CM	CREATE.inv
target-mo	—	by UM	CREATE.inv
target-action-id	—	by UM	CREATE.inv
action-parameter-pairs	—	by UM	CREATE.inv
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

5.11.6 ACTIONS—ProxyAction MO CLASS

There are no actions defined for this managed object.

5.11.7 STATE-RELATED BEHAVIOR—ProxyAction MO CLASS

This managed object class has no state-related behavior.

5.11.8 NOTIFICATIONS—ProxyAction MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the SLEManagementEntity managed object class.

5.11.9 VALIDATION ASPECTS—ProxyAction CLASS

a) The managed objects specified as targets must exist. If not, a 'completeness defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

b) At validation time, the parameters specified in the *action-parameter-pairs* must match the action-parameters defined for their respective actions. If not, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

5.12 ProxySet MANAGED OBJECT CLASS

5.12.1 PURPOSE

A proxySet managed object specifies a target managed object and a SET operation that is to be invoked on that target object when the triggering event or failure condition associated with the containing eventHandler managed object occurs. These triggering events and failure conditions are specified as part of the EventHandler managed object (5.6) that contains the proxySet managed object.

5.12.2 INHERITANCE—ProxySet MO CLASS

This managed object class is derived from and inherits the attributes of the SLEManagementEntity managed object class.

5.12.3 OBJECTS CONTAINED—ProxySet CLASS

An instance of this concrete managed object class contains no other managed objects.

5.12.4 STATES—ProxySet MO CLASS

The ProxySet managed object has no states.

5.12.5 ATTRIBUTES—ProxySet MO CLASS

This managed object class has the following specific attributes:

a) **proxy-set-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (see annex C).

   1) Data type/values: Character string.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **target-mo.** The target managed object, i.e., the managed object affected by the SET operation.

   1) Data type/values: DN.

2) Modifications: SET shall be enabled when the containing eventHandler managed object is in the state *waiting*.

c) **attribute-value-pairs.** A set of (*attribute-id*, *value*) data structures, where attribute-id is the identifier of an attribute and the value is taken from the data type and range defined for that attribute.

1) Data type/values: See 3.4.2.2.3(d).

2) Modifications: SET shall be enabled when the containing eventHandler managed object is in the state *waiting*.

Refer to table 5-11 for the proxySet managed object attribute summary.

Table 5-11: Attribute Summary—ProxySet Managed Object

Attribute	Inherited From	Initial Value	Specified In
proxy-set-mo-id	—	by CM	CREATE.inv
target-mo	—	by UM	CREATE.inv
attribute-value-pairs	—	by UM	CREATE.inv
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

5.12.6 ACTIONS—ProxySet MO CLASS

There are no actions defined for this managed object.

5.12.7 STATE-RELATED BEHAVIOR—ProxySet MO CLASS

This managed object class has no state-related behavior.

5.12.8 NOTIFICATIONS—ProxySet MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the SLEManagementEntity managed object class.

5.12.9 VALIDATION ASPECTS—ProxySet CLASS

a) The managed objects specified as targets must exist at validation time. If not, a ‘completeness defect’ shall be added to the containing servicePackage managed object’s set of *defect-records*.

b) The attributes specified in the *attribute-value-pairs* must match the attributes defined for their respective targets at validation time. If not, a ‘consistency defect’ shall be added to the containing servicePackage managed object’s set of *defect-records*.

5.13 R-ocf-ts MANAGED OBJECT CLASS

5.13.1 PURPOSE

This managed object represents the resources to provide and use Operational Control Field (OCF) transfer services. See 5.20 for a general discussion of transfer service provision and use.

5.13.2 INHERITANCE—R-ocf-ts MO CLASS

This managed object class is derived from and inherits the properties of the R-transferService managed object class.

5.13.3 OBJECTS CONTAINED—R-ocf-ts MO CLASS

NOTE – This managed object class is an abstract class; thus, it cannot be instantiated and containment relationships do not apply.

5.13.4 STATES—R-ocf-ts MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the R-transferService managed object class.

5.13.5 ATTRIBUTES—R-ocf-ts MO CLASS

This managed object class has the following specific attributes:

a) **ocf-type.** The type of OCFs that will be transferred to the user.

   1) Data type/values:

      i) 'all-ocf-types' indicates that all ocf types are to be sent;

      ii) 'only-ocf-type-1' indicates that only CLCW reports are to be sent;

      iii) 'only-ocf-type-2-x' indicates that only non-CLCW reports are to be sent;

      iv) 'only-ocf-type-2-0' indicates that only project-specific OCFs are to be sent;

      v) 'only-ocf-type-2-1' indicates that only reserved OCFs are to be sent.

      NOTE – OCF type 2-1 has not yet been defined by CCSDS.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

b) **allowed-ocf-update-mode.** The set of acceptable values that the OCF update mode can assume.

   1) Data type/values:

i) ‘continuous’ if the user is allowed to select the continuous update mode;

ii) ‘change-based’ of the user is allowed to select the change based mode;

iii) ‘time-based’ if the user is allowed to select the time based mode;

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **current-ocf-update-mode.** The update mode.

1) Data type/values:

i) ‘continuous’ if all OCFs meeting the selection criteria are to be sent, including changes and duplications;

ii) ‘change-based’ if the OCFs meeting the selection criteria are to be sent only if the value of the OCF has changed;

iii) ‘time-based’ if the OCFs meeting the selection criteria are to be sent periodically;

iv) ‘undefined’ if the service instance has not yet been started.

2) Modifications: SET shall be disabled. SLE Complex Management may modify this attribute when the managed object is in the state *active*. An R-OCF service provider entity shall update this attribute upon successfully performing a START invocation. An R-OCF service user entity shall update this attribute upon the return of a positive result for the START invocation.

d) **allowed-tc-vc.** This attribute defines acceptable values for the telecommand virtual channel with which the Command Link Control Word (CLCW) is to be associated.

1) Data type/values: A set of unsigned integers between 0 and 63.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

e) **current-tc-vc.** This attribute identifies the telecommand virtual channel that is currently in effect for the service instance, as specified in the most recent successful START invocation for the service instance.

1) Data type/values:

i) ‘undefined’ if the service instance has not yet been started;

ii) unsigned integer in the range [0..63].

2) Modifications: SET shall be disabled. An R-OCF service provider entity shall update this attribute upon successfully performing a START invocation. An R-OCF service user entity shall update this attribute upon the return of a positive result for the START invocation.

f) **allowed-ocf-channel-dns.** The set of DNs of the OCF data channels that may be selected by the user for transfer via the associated R-OCF service instance. Each DN must be formed according to the rules defined in annex C.

   1) Data type/values: DN.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

Refer to table 5-12 for the r-ocf-ts managed object attribute value summary.

Table 5-12: Attribute Value Summary—R-ocf-ts Managed Object

Attribute	Inherited From	Initial Value	Specified In
ocf-type	—	by UM	CREATE.inv
allowed-ocf-update-mode	—	by UM	CREATE.inv
current-ocf-update-mode	—	= ‘undefined’	No
allowed-tc-vc	—	by UM	CREATE.inv
current-tc-vc	—	= ‘undefined’	No
allowed-ocf-channel-dns	—	by UM	CREATE.inv
r-delivery-mode	R-transferService (5.15)	by UM	CREATE.inv
delivery-quality	R-transferService (5.15)	by UM	CREATE.inv
transfer-buffer-size	R-transferService (5.15)	by UM	CREATE.inv
latency-limit	R-transferService (5.15)	by UM	CREATE.inv
max-number-concatenations	R-transferService (5.15)	by UM	CREATE.inv
current-ts-version	—	by UM	CREATE.inv
lower-bound-reporting-period	—	by UM	CREATE.inv
time-out-period	—	by UM	CREATE.inv
transfer-service-state	—	= *waiting*	—
association-initiation-responsibility	—	by UM	CREATE.inv
responder-port-identifier	—	by UM	CREATE.inv
local-sle-entity-identifier	—	by CM	—
peer-sle-entity-access-list	—	by UM	CREATE.inv
scheduled-period-start-time	—	by UM	CREATE.inv
scheduled-period-stop-time	—	by UM	CREATE.inv
first-data-unit-time	—	by CM	when first data unit is transferred
last-data-unit-time	—	by CM	when last data unit is transferred
service-instance-start	—	= ‘null’	when first service instance starts
service-instance-stop	—	= ‘null’	when last service instance stops
provision-time	—	= 0	as needed
failed-binds	—	= 0	as needed
unavailable-time	—	= 0	as needed
done-time	—	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

5.13.6 ACTIONS—R-ocf-ts MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the R-transferService managed object class.

5.13.7 STATE-RELATED BEHAVIOR—R-ocf-ts MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the R-transferService managed object class.

5.13.8 NOTIFICATIONS—R-ocf-ts MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the R-transferService managed object class.

5.13.9 VALIDATION ASPECTS—R-ocf-ts MO CLASS

a) The *allowed-ocf-channel-dns* must exist in the channel tree and must match the channel type that the present managed object can process. If the respective channel cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

b) For each ocf channel included in the *allowed-ocf-channel-dns* attribute, the set of values contained in the *ocf-type-options* attribute of the corresponding r-ocf-channel managed object must include the value of the *ocf-type* attribute of this r-ocf-ts managed object.

5.14 R-ocf-ts-u MANAGED OBJECT CLASS

5.14.1 PURPOSE

This managed object represents the resources needed to use a return OCF transfer service. The R-ocf-ts-u managed object class is needed to support the Command Operation Procedure (COP) processing with access to an OCF transfer provided by another SLE Complex. See 5.20 for a general discussion of transfer service use.

5.14.2 INHERITANCE—R-ocf-ts-u MO CLASS

This managed object class is derived from and inherits the properties of the R-ocf-ts managed object class.

5.14.3 OBJECTS CONTAINED—R-ocf-ts-u MO CLASS

An instance of this concrete managed object class contains no other managed objects.

5.14.4 STATES—R-ocf-ts-u MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the R-ocf-ts managed object class.

5.14.5 ATTRIBUTES—R-ocf-ts-u MO CLASS

This managed object class has the following specific attributes:

a) **r-ocf-ts-u-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (see annex C).

   1) Data type/value: Character string with a maximum length of 256 ASCII characters.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **service-instance-identifier.** The service instance to be used.

   1) Data type/value: The DN of the transfer 'service provided' managed object within the service-providing SLE Complex associated with the desired service instance.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **OCF-data-units-consumed**. The number of OCF channel data units consumed by the service instance.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

d) **OCF-PDUs-consumed**. The number of OCF-PDUs consumed by the service instance.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase, as follows:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

e) **ert-interval-begin**. The value to be placed in the start-time parameter of the START invocation.

   1) Data type/values: CCSDS time code.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

   3) Dependencies: This attribute is applicable only if the value of *r-delivery-mode* is 'return offline'.

f) **ert-interval-end**. The value to be placed in the stop-time parameter of the START invocation.

   1) Data type/values: CCSDS time code.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

   3) Dependencies: This attribute is applicable only if the value of *r-delivery-mode* is ‘return offline’.

g) **initial-ocf-update-mode**. The update mode that is to be specified in the START invocation that is generated at the *scheduled-period-start-time* by the SLE Complex that is using the OCF transfer service instance.

   1) Data type/values:

      i) ‘continuous’ if the *ocf-update-mode* is to be set to ‘continuous’ in the START invocation;

      ii) ‘change-based’ if the *ocf-update-mode* is to be set to ‘change-based’ in the START invocation;

      iii) ‘time-based’ if the *ocf-update-mode* is to be set to ‘time-based’ in the START invocation.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

h) **initial-tm-vc**. The TM VC that is to be specified in the START invocation that is generated at the *scheduled-period-start-time* by the SLE Complex that is using the OCF transfer service instance.

   1) Data type/values: Global Virtual Channel Identifiers (GVCID) corresponding to the set of OCF channels contained in the attribute *allowed-ocf-channel-dns*.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

Refer to table 5-13 for the r-ocf-ts-u managed object attribute value summary.

Table 5-13: Attribute Value Summary—R-ocf-ts-u Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-ocf-ts-u-mo-id	—	by CM	CREATE.inv
service-instance-identifier	—	by UM	CREATE.inv
ocf-data-units-consumed	—	= 0	as needed
ocf-pdus-consumed	—	= 0	as needed
ert-interval-begin	—	by UM	CREATE.inv
ert-interval-end	—	by UM	CREATE.inv
initial-ocf-update-mode	—	by UM	CREATE.inv
initial-tm-vc	—	by UM	CREATE.inv
ocf-type	R-ocf-ts (5.13)	by UM	CREATE.inv
allowed-ocf-update-mode	R-ocf-ts (5.13)	by UM	CREATE.inv
allowed-tc-vc	R-ocf-ts (5.13)	by UM	CREATE.inv
initial-ocf-type	R-ocf-ts (5.13)	by UM	CREATE.inv
initial-ocf-update-mode	R-ocf-ts (5.13)	by UM	CREATE.inv
initial-tc-vc	R-ocf-ts (5.13)	by UM	CREATE.inv
allowed-sle-channel-dn	R-ocf-ts (5.13)	by UM	CREATE.inv
r-delivery-mode	R-transferService (5.15)	by UM	CREATE.inv
delivery-quality	R-transferService (5.15)	by UM	CREATE.inv
transfer-buffer-size	R-transferService (5.15)	by UM	CREATE.inv
latency-limit	R-transferService (5.15)	by UM	CREATE.inv
max-number-concatenations	R-transferService (5.15)	by UM	CREATE.inv
current-ts-version	—	by UM	CREATE.inv
lower-bound-reporting-period	—	by UM	CREATE.inv
time-out-period	—	by UM	CREATE.inv
transfer-service-state	—	= *waiting*	—
association-initiation-responsibility	—	by UM	CREATE.inv
responder-port-identifier	—	by UM	CREATE.inv
local-sle-entity-identifier	—	by CM	—
peer-sle-entity-access-list	—	by UM	CREATE.inv
scheduled-period-start-time	—	by UM	CREATE.inv
scheduled-period-stop-time	—	by UM	CREATE.inv
first-data-unit-time	—	by CM	when first data unit is transferred
last-data-unit-time	—	by CM	when last data unit is transferred
service-instance-start	—	= ‘null’	when first service instance starts
service-instance-stop	—	= ‘null’	when last service instance stops
provision-time	—	= 0	as needed
failed-binds	—	= 0	as needed
unavailable-time	—	= 0	as needed
done-time	—	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

5.14.6 ACTIONS—R-ocf-ts-u MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the R-ocf-ts managed object class.

5.14.7 STATE-RELATED BEHAVIOR—R-ocf-ts-u MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the R-ocf-ts managed object class.

5.14.8 NOTIFICATIONS—R-ocf-ts-u MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the R-ocf-ts managed object class.

5.14.9 VALIDATION ASPECTS—R-ocf-ts-u MO CLASS

a) The name of the channel used shall be appropriate for an instance of return operational control field service.

b) The *allowed-sle-channel-dn* must exist in the channel tree and must match the channel type that the present managed object can process. If the channel cannot be found or if there is a channel type mismatch, a ‘consistency defect’ shall be added to the containing servicePackage managed object’s set of *defect-records*.

5.15 R-transferService MANAGED OBJECT CLASS

5.15.1 PURPOSE

This managed object represents the resources needed to either provide or use return transfer services. It inherits the properties of its parent TransferService managed object class. Refer to 5.20 for a general discussion of transfer service provision and use.

5.15.2 INHERITANCE—R-transferService MO CLASS

This managed object class is derived from and inherits the properties of the TransferService managed object class.

5.15.3 OBJECTS CONTAINED—R-transferService MO CLASS

This managed object class is an abstract class; thus it cannot be instantiated and containment relationships do not apply.

5.15.4 STATES—R-transferService MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the TransferService managed object class.

5.15.5 ATTRIBUTES—R-transferService MO CLASS

This managed object class has the following specific attributes:

a) **r-delivery-mode.** The *delivery-mode* as defined in the *Cross Support Reference Model* (reference [1]) and the SLE Service Specifications (references [2]-[8]).

   1) Data type/values:

      i) 'return online';

      ii) 'return offline'.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

b) **delivery-quality.** The *delivery-quality* as defined in references [1] through [8].

   1) Data type/values:

      i) 'timely';

      ii) 'complete';

      iii) 'null'—this value must be specified if and only if *r-delivery-mode* is 'return offline'.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **transfer-buffer-size.** The size (in number of maximum-length transfer service TRANSFER-DATA operation messages) for the transfer buffer.

   1) Data type/values: Unsigned integer. The minimum shall be 200 maximum-length transfer service TRANSFER-DATA operation messages.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

   3) Dependencies: The value of this attribute depends on which transfer service is under consideration.

d) **latency-limit.** The *latency-limit* is the maximum delay (in seconds) between reception of an SL-DU and the delivery of the associated SLE-SDU to the user.

   1) Data type/values: Unsigned integer.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

3) Dependencies: This attribute only influences SLE-SDU provision if the service instance delivery mode is timely online. If the service instance delivery mode is complete online or offline, this attribute does not influence SLE-SDU provision.

e) **max-number-concatenations**. The maximum concatenations to a single unit in the transfer buffer. When this limit is reached, the transfer buffer is released for transmission.

1) Data type/values: Unsigned integer.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

3) Dependencies: The meaning of this attribute depends on the value of *r-delivery-mode*:

i) If *r-delivery-mode* is 'return online', this attribute defines the maximum number of service TRANSFER-DATA and transfer service SYNC-NOTIFY invocations to be concatenated in a single unit in the transfer buffer.

ii) If *r-delivery-mode* is 'return offline', this attribute defines the maximum number of transfer service TRANSFER-DATA invocations to be concatenated in a single unit in the transfer buffer.

NOTE – If the value of *delivery-quality* is 'timely', the actual number of invocations in the transfer buffer when it is released may be less than the maximum value allowed, due to the latency limit being reached.

Refer to table 5-14 for the r-transferService managed object attribute value summary.

Table 5-14: Attribute Value Summary—R-transferService Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-delivery-mode	—	by UM	CREATE.inv
delivery-quality	—	by UM	CREATE.inv
transfer-buffer-size	—	by UM	CREATE.inv
latency-limit	—	by UM	CREATE.inv
max-number-concatenations	—	by UM	CREATE.inv
current-ts-version	TransferService (5.20)	by UM	CREATE.inv
lower-bound-reporting-period	TransferService (5.20)	by UM	CREATE.inv
time-out-period	TransferService (5.20)	by UM	CREATE.inv
transfer-service-state	TransferService (5.20)	= *waiting*	no
association-initiation-responsibility	TransferService (5.20)	by UM	CREATE.inv
responder-port-identifier	TransferService (5.20)	by UM	CREATE.inv
local-sle-identifier	TransferService (5.20)	by CM	—
peer-sle-entity-access-list	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-start-time	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-stop-time	TransferService (5.20)	by UM	CREATE.inv
first-data-unit-time	TransferService (5.20)	by CM	as needed
last-data-unit-time	TransferService (5.20)	by CM	as needed
service-instance-start	TransferService (5.20)	by CM	as needed
service-instance-stop	TransferService (5.20)	by CM	as needed
provision-time	TransferService (5.20)	by CM	as needed
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

5.15.6 ACTIONS—R-transferService MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the TransferService managed object class.

5.15.7 STATE-RELATED BEHAVIOR—R-transferService MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the TransferService managed object class.

5.15.8 NOTIFICATIONS—R-transferService MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the TransferService managed object class.

5.15.9 VALIDATION ASPECTS—R-transferService MO CLASS

No specific validation aspects are defined for this managed object class. However, validation aspects are inherited from the TransferService managed object class.

5.16 ServiceAgreement MANAGED OBJECT CLASS

5.16.1 PURPOSE

5.16.1.1 General

a) A Service Agreement is concluded between suitable organizational units in an MDOS and in a Complex. It may cover many different aspects of cross support. With respect to the SLE Service Management interaction, a Service Agreement establishes two kinds of information:

   1) information that enables management interaction between an MDOS and an SLE Complex;

   2) information that is needed during the management interaction between MDOS and SLE Complex.

b) The first kind of information includes:

   1) the version of the Recommendation to be used for management interactions;

   2) the profile of the management port;

   3) selection of a ground station to support the Space Link session (only if the SLE Complex performs Space Link acquisition);

   4) the technology chosen for implementation of the management port, and the respective Recommendation for the mapping rules and gateway, if required;

   5) a list of services, production, and data storage to be performed;

   6) the logical address of the management port;

   7) the DN of the serviceAgreement managed object(s) that are exposed at the management port according to annex C of this draft Recommendation;

   8) the security measures (e.g. authentication, encryption) to be applied to the communication between SLE Utilization Management and SLE Complex Management and the respective security information items;

   9) the identification of the performer of the NOTIFY operation in the domain of SLE Utilization Management.

   NOTES

   1 Recall from 3.4.2.5 that the NOTIFY operation causes SLE Utilization Management to receive and process a notification. Thus, SLE Utilization Management is the performer and SLE Complex Management is the invoker. Each managed object definition specifies the notifications that the managed object may issue.

2 This draft Recommendation assumes that for a given implementation technology, mapping rules exist that ensure there is an entity on the side of SLE Utilization Management that performs the NOTIFY operation.

c) All information items above need bilateral agreement prior to any use of the management operations, which are defined in section 3 of this draft Recommendation.

d) The second kind of information is specified in the ServiceAgreement managed object class. It covers information items that are valid for a number of servicePackage managed objects and for their contained managed objects during their lifetimes.

5.16.1.2 Perception by SLE Utilization Management

a) The Service Agreement specifies properties of the Complex Management Port required for monitoring and controlling services required from a given Complex during a specific mission phase.

b) Service Agreements have disjoint Service Agreement periods. For a given mission and spacecraft at a given point in time, either no serviceAgreement managed object is *applicable* or a single serviceAgreement managed object is *applicable*.

NOTES

1 In order to support different phases of a specific mission (e.g., Launch and Early Orbit phase, Descent phase, Hibernation phase), SLE Utilization Management may establish several Service Agreements with a given SLE Complex Management. In this case, it may be possible that some Service Agreements have been successfully negotiated and are considered binding for both parties. SLE Utilization Management may request SLE Complex Management to expose an established serviceAgreement managed object at the Complex Management port.

2 The establishment of an SLE Service Agreement may take place under the umbrella of a contract of other formal arrangement that defines the two organizations' legal and financial responsibilities. The form and method of developing or administering such a contract is outside the scope of this draft Recommendation. Only the technical aspect of the agreement, as reflected in the serviceAgreement managed object, is relevant to this draft Recommendation.

3 It is possible to alternate between two Service Agreements (e.g., near earth and deep space positions of a highly elliptical orbit).

5.16.1.3 Perception by SLE Complex Management

a) A Service Agreement constrains the range of attribute values of the exposed managed objects and allows safe sharing of Complex resources between different missions.

b) SLE Complex Management shall ensure that the negotiated attribute values in a Service Agreement are consistent with the Complex's capabilities that are available (or will become available) during the prospective Service Agreement period.

NOTE – The means and procedures for the conduct of such negotiations are outside the scope of this draft Recommendation.

5.16.1.4 Security Aspects

Only managed objects contained (at any level) in a serviceAgreement managed object are exposed to SLE Utilization Management.

5.16.1.5 Creation Aspects

a) Unlike all other managed objects, a serviceAgreement managed object is not created by SLE Utilization Management; instead, SLE Complex Management exposes it at the Complex Management Port as the first in the set of exposed managed objects.

b) The means by which a serviceAgreement managed object is exposed by SLE Complex Management are outside the scope of this draft Recommendation.

c) Once SLE Complex Management and SLE Utilization Management have negotiated and agreed upon all attribute values, the serviceAgreement managed object is ready to be exposed.

d) Once exposed, a serviceAgreement managed object may be augmented with contained managed objects.

e) The means by which the applicable value of the naming attribute and the related security information are conveyed to SLE Utilization Management are outside the scope of this draft Recommendation.

5.16.1.6 Deletion Aspects

a) SLE Utilization Management cannot request the deletion of a serviceAgreement managed object.

b) Deletion of a serviceAgreement managed object is outside the scope of this draft Recommendation.

5.16.2 INHERITANCE—ServiceAgreement MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

5.16.3 OBJECTS CONTAINED—ServiceAgreement MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero or one spaceLinkPhysicalLayerCharacteristics managed object. Presence of an instance is mandatory if the Complex operates the space link. This class is specified in reference [13].

b) One dataChannelProfile managed object. Presence of an instance is mandatory. This class is specified in 6.1.

c) Zero, one or more dataStore managed objects. Presence of an instance mandatory if the Complex provides transfer services that include data storage and possible subsequent retrieval. This class is specified in 5.4.

d) One or more servicePackage managed objects. Presence of an instance is mandatory. This class is specified in 5.17.

e) One authenticationInformationBase managed object. Presence of an instance is mandatory. This class is specified in 5.1.

f) One telecommunicationInformationBase managed object. Presence of an instance is mandatory. This class is specified in 5.19.

5.16.4 STATES—ServiceAgreement MO CLASS

A serviceAgreement managed object shall exist in one of the following states:

a) **Agreed.** The Service Agreement has been successfully arranged. The serviceAgreement managed object has been exposed at the Complex Management interface.

b) **Applicable.** The Service Agreement period has begun.

5.16.5 ATTRIBUTES—ServiceAgreement MO CLASS

NOTE – The values for this managed object's attributes are assigned using a negotiation process that is outside the scope of this draft Recommendation.

This managed object class has the following specific attributes:

a) **service-agreement-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (see annex C).

   1) Data type/values: Character string.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

b) **contractual-reference**. The set of contractually binding documents, which together shall provide the framework for the cross support activity and the enabling information for interaction between SLE Utilization Management and SLE Complex Management.

   1) Data type/values: Set of character strings.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

c) **service-agreement-start**. The date and time from which the Service Agreement is considered *applicable.*

   1) Data type/values: CCSDS time code.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

d) **service-agreement-stop**. The date and time that the Service Agreement period ends.

   1) Data type/values: CCSDS time code.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

e) **max-num-data-store**. The maximum number of dataStore managed objects that may be contained by this serviceAgreement.

   1) Data type/values: Unsigned integer.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

f) **max-accumulated-size**. The maximum accumulated size (in GBytes) of storage resources that can be assigned to SLE Utilization Management during this Service Agreement period.

   1) Data type/values: Unsigned integer.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

g) **max-serv-pack**. The maximum number of servicePackage managed objects that may be contained by this serviceAgreement managed object at a given time.

   1) Data type/values: Unsigned integer.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

h) **max-notification-log-size**. The maximum number of recorded notifications.

   1) Data type/values: Integer. If the value specified is ‘not used’, notifications shall not be recorded when a telecommunication association for immediate transfer does not exist.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

i) **notification-timeout**. This specifies a period of time (in seconds) during which a managed object suppresses subsequent notifications of a given type after a notification has been issued. The behavior is specified in 4.9.2.

   1) Data type/values: Integer.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

j) **trajectory-filestore-id**. The directory of trajectory prediction information files in the realm of SLE Complex Management.

   1) Data type/values: The attribute value shall be expressed as a Uniform Resource Name (URN), as specified in *Uniform Resource Names* (reference [14]), that can be resolved by the supporting SLE Complex.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

k) **max-size-trajectory-filestore**. The maximum size of trajectory prediction file (in Mbytes).

   1) Data type/values: Unsigned real number.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

l) **trajectory-format**. The format of the trajectory prediction file mutually agreed upon between SLE Complex Management and SLE Utilization Management.

   1) Data type/values: Character string.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

   3) Dependencies: This attribute is mandatory if the Complex performs Space Link acquisition.

m) **anonymity-flag**. The identity of the mission is/is not to be revealed.

   1) Data type/values: The default value shall be 'anonymity not required'. This shall have the value 'anonymity required' if the identity of the mission is not to be revealed.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

n) **event-list**. A set of (*event-identifier, description*) pairs. Each pair is associated with an event that may occur at a Complex and may be of particular interest to SLE Utilization Management.

   1) Data type/value: Each pair shall be made up of:

      i) **event-identifier.** This identifies the event by an integer.

      ii) **description.** The meaning of the associated event described in free text or a reference to the event's description.

2) Modifications: This attribute shall not be modifiable using the SET operation.

3) Dependencies: If an eventHandler managed object is contained in one of a given serviceAgreement's servicePackages, this attribute shall be mandatory and the set must contain at least one member.

NOTES

1 THROW-EVENT operations of transfer Services have an 'event-identification' parameter, which identifies an event caused by a Complex-external entity. It is recommended that these 'event-identification' parameters are directly used as *event-identifier* in an appropriate (*event-identifier, description*) pair.

2 The precise meaning that a Complex Management associates with *event-identifiers* other than those reserved for THROW-EVENT support is not specified by this draft Recommendation.

3 For a given *event-identifier*, SLE Utilization Management may create an eventHandler managed object that reacts to the occurrence of that specific event.

4 The means by which the occurrence of a Complex-internal event of a THROW-EVENT invocation is realized and linked to a related eventHandler managed object are not specified by this draft Recommendation.

5 The agreed set of (*event-identifier, description*) pairs is part of the binding information (see 3.3.4). The means to achieve this agreement are outside the scope of this draft Recommendation.

o) **max-recovery-timeout.** The maximum time period in seconds, after which a corruption of Complex resources is considered unrecoverable (this is sometimes referred to as an 'Unrecoverable Failure').

1) Data type/values: Character string.

2) Modifications: This attribute shall not be modifiable using the SET operation.

p) **service-agreement-state.** The current state of the ServiceAgreement managed object.

1) Data type/values: See 5.16.4.

2) Modifications: This attribute shall not be modifiable using the SET operation. See 5.16.7.

Refer to table 5-15 for the serviceAgreement managed object attribute value summary.

Table 5-15: Attribute Value Summary—ServiceAgreement Managed Object

Attribute	Inherited From	Initial Value	Specified In
service-agreement-mo-id	—	by CM	—
contractual-reference	—	by UM	—
service-agreement-start	—	—	—
service-agreement-stop	—	—	—
max-num-data-store	—	—	—
max-accumulated-size	—	—	—
max-serv-pack	—	—	—
max-notification-log-size	—	—	—
max-notification-threshold	—	by CM	—
notification-timeout	—	by CM	—
notification-suppressed-period	—	by CM	—
trajectory-filestore-id	—	—	—
max-size-trajectory-filestore	—	—	—
trajectory-format	—	—	—
anonymity-flag	—	= anonymity not required	—
event-list	—	by UM	—
max-recovery-timeout	—	by CM	—
service-agreement-state	—	= agreed	—
read-me	SLEManagementEntity (5.18)	by UM	—

5.16.6 ACTIONS—ServiceAgreement MO CLASS

A serviceAgreement managed object performs no actions.

NOTE – However, objects contained in a serviceAgreement may perform actions.

5.16.7 STATE-RELATED BEHAVIOR—ServiceAgreement MO CLASS

5.16.7.1 General

See figure 5-6 for the serviceAgreement state transition diagram.

NOTES

1 SLE Utilization Management may simultaneously negotiate several Service Agreements with a given Complex for the same mission and the same spacecraft. In this case, it may be possible that some Service Agreements have been successfully negotiated and are considered binding for both parties.

2 In order to support different phases of a specific mission, this draft Recommendation assumes that SLE Utilization Management will establish several Service Agreements with a given SLE Complex Management, one for each phase.

3 GET operation requests are accepted in states agreed or applicable.

4 No general rules are defined in this subsection for the acceptance or rejection of SET operation requests. They are state- and attribute-dependent and, hence, are part of the attributes description.

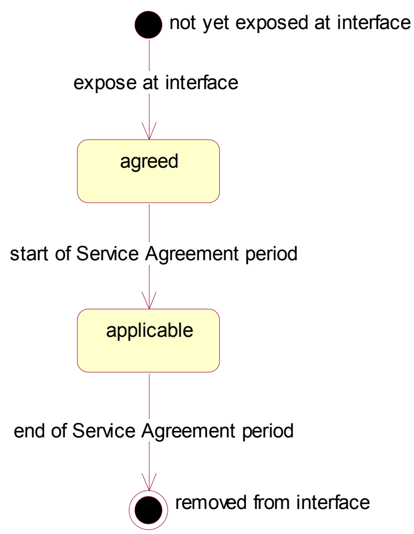


Figure 5-6: ServiceAgreement State Transition Diagram

5.16.7.2 AGREED State

5.16.7.2.1 General

When in the state *agreed*, the serviceAgreement managed object:

a) can be declared *applicable* by SLE Complex Management;

b) is used as the basis for the validation process;

c) enables the allocation of Complex resources for use by the mission and spacecraft for the period in which the serviceAgreement managed object is *applicable*.

5.16.7.2.2 Conditions for the serviceAgreement managed object entering the state *agreed*

If it is exposed at the Complex management port.

5.16.7.2.3 Operations accepted and rejected by the serviceAgreement managed object in the state *agreed*

Accepts a CREATE invocation for contained managed objects.

5.16.7.3 APPLICABLE State

5.16.7.3.1 General

When in the state *applicable*, the serviceAgreement managed object:

a) is used as the basis for the validation process;

b) enables the allocation of Complex resources for use by the mission and spacecraft.

5.16.7.3.2 Conditions for the serviceAgreement managed object entering the state *applicable*

At the start of the Service Agreement period.

5.16.7.3.3 Operations accepted and rejected by the serviceAgreement managed object in the state *agreed*

Accepts a CREATE invocation for contained managed objects.

5.16.8 NOTIFICATIONS—ServiceAgreement MO CLASS

Although a serviceAgreement managed object issues no notifications, objects contained in a serviceAgreement may issue their own notifications.

5.16.9 VALIDATION ASPECTS—ServiceAgreement MO CLASS

Information in the serviceAgreement is used to validate the service package and its contained objects.

5.17 ServicePackage MANAGED OBJECT CLASS

5.17.1 PURPOSE

5.17.1.1 General

ServicePackage is a concrete managed object class that defines a collection of SLE transfer services for the SLE Complex to provide to the space mission's MDOS. Contained managed objects represent these transfer services. Within the limitations provided in the SLE Service Agreement, SLE Utilization Management coordinates with SLE Complex Management to create instances of the ServicePackage managed object class. The SLE service package's contained objects' attribute values represent the schedule for providing SLE transfer services.

An expiration time for the SLE service package itself is an attribute of the servicePackage managed object.

5.17.1.2 Perception by SLE Utilization Management

a) A servicePackage is a product of a mission planning function. A servicePackage shall cover the support needs of this mission with respect to a single Complex.

NOTE – If a single Complex cannot fully satisfy the mission’s service requirements, SLE Utilization Management may seek cooperation with other Complexes. If suitable Complexes are identified (and Service Agreements are concluded), SLE Utilization Management prepares specific service packages for these Complexes. SLE Utilization Management coordinates these service packages.

b) SLE Utilization Management shall submit a service package to SLE Complex Management by requesting the creation of a servicePackage managed object.

c) If a service package is complete (meaning that all components have been specified by creating contained managed objects), SLE Utilization Management shall request its validation. See 5.16.6 for a discussion of the *validate* action.

d) If a service package’s validation failed, SLE Utilization Management should:

   1) inspect the set of *defect-records*;

   2) revise and modify the service package.

e) In particular, if a conflict is identified, SLE Utilization Management should:

   1) contact SLE Utilization Management of other missions that submitted the conflicting service packages;

   2) negotiate with the other SLE Utilization Management concerned and agree on measures to resolve the conflict;

   3) modify its own servicePackage as necessary;

   4) request a *validate* action.

   NOTES

   1 It is assumed that these negotiations require human involvement.

   2 In order to preserve anonymity, it is conceivable that SLE Complex Management interact only via a human mediator, e.g., the Complex Manager.

   3 The means by which the necessary contacts are established and the way in which the negotiations are performed are not covered by this draft Recommendation.

4 To resolve a conflict, it may be necessary for several SLE Utilization Managements to revise their servicePackages and synchronize their re-validation requests.

5 It may happen that servicePackages contained by the same serviceAgreement managed object are in conflict.

5.17.1.3 Perception by SLE Complex Management

SLE Complex Management shall be responsible for:

a) analyzing each service package in a validation process, which covers conformance with the serviceAgreement, completeness, internal consistency, and conflicts with other servicePackage managed objects.

b) informing SLE Utilization Management of any defects detected during the validation process. SLE Complex Management shall NOT attempt to resolve the defects on its own.

5.17.2 INHERITANCE—ServicePackage MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

5.17.3 OBJECTS CONTAINED—ServicePackage MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero or one antennaPointing managed object. Presence of an instance is mandatory if the Complex performs space link acquisition. This class is specified in 5.1.

b) Zero or one eventControl managed object. This class is specified in 5.5.

c) Zero or one notificationLog managed object. Presence of an instance indicates that notifications shall be logged when there is no telecommunication association with SLE Utilization Management. This class is specified in 5.9.

d) Zero, one or more instances of a return technology-specific SpaceLinkCarrierPackage derived class.

e) Zero, one or more instances of a forward technology-specific SpaceLinkCarrier-Package derived class.

NOTE – Items (d) and (e) are defined in *Physical Layer*, reference [13].

f) Zero, one or more f-bitstream-st managed objects. This class is specified in 14.1.

g) Zero, one or more f-cltu-st managed objects. At least one instance is mandatory if the Complex provides offline FCLTU service. This class is specified in 10.1.

h) Zero, one or more f-cltu-ts-u managed objects. This class is specified in 11.2.

i) Zero, one or more f-c/vcdu-st managed objects. This class is specified in 13.2.

j) Zero, one or more f-insert-st managed objects. This class is specified in 13.4.

k) Zero, one or more f-proto-vcdu-st managed objects. At least one instance is mandatory if the Complex provides offline protoVCDU service. This class is specified in 14.5.

l) Zero, one or more f-proto-vcdu-ts-u managed objects. This class is specified in 14.7.

m) Zero, one or more f-sp-st managed objects. At least one instance is mandatory if the Complex provides offline FSP service. This class is specified in 12.1.

n) Zero, one or more f-tc-f-st managed objects. At least one instance is mandatory if the Complex provides offline FTCF service. This class is specified in 11.3.

o) Zero, one or more f-tc-f-ts-u managed objects. This class is specified in 12.3.

p) Zero, one or more f-tc-vca-st managed objects. At least one instance is mandatory if the Complex provides offline TCVCA service. This class is specified in 12.5.

q) Zero, one or more f-vca-st managed objects. This class is specified in 14.8.

r) Zero, one or more r-af-st managed objects. At least one instance is mandatory if the Complex provides offline RAF service. This class is specified in 7.2.

s) Zero, one or more r-af-ts-u managed objects. This class is specified in 8.1.

t) Zero, one or more r-bitstream-st managed objects. At least one instance is mandatory if the Complex provides offline r-bitstream service. This class is specified in 9.1.

u) Zero, one or more r-cf-st managed objects. At least one instance is mandatory if the Complex provides offline RCF service. This class is specified in 8.3.

v) Zero, one or more r-cf-ts-u managed objects. This class is specified in 9.3.

w) Zero, one or more r-fsh-st managed objects. At least one instance is mandatory if the Complex provides offline Frame Secondary Header (FSH) service. This class is specified in 8.6.

x) Zero, one or more r-insert-st managed objects. At least one instance is mandatory if the Complex provides offline r-insert service. This class is specified in 7.5.

y) Zero, one or more r-ocf-st managed objects. This class is specified in 8.9.

z) Zero, one or more r-ocf-ts-u managed objects. This class is specified in 5.14. An instance must be contained if:

   1) the service package contains either an f-tc-session-prod or f-tc-vc-prod managed object; and

2) the service package does not contain an instance of R-cf-prod.

aa) Zero, one or more r-sp-st managed objects. At least one instance is mandatory if the complex provides offline RSP service. This class is specified in 9.5.

5.17.4 STATES—ServicePackage MO CLASS

a) In the definition phase, a servicePackage managed object shall exist in one of the following states:

1) **Waiting.** The managed object is created and may be edited or submitted to the validation process.

2) **Under validation.** The validation of the servicePackage managed object is in progress. This state can be entered during the definition phase of the servicePackage. All contained managed objects are static during servicePackage managed object validation.

b) In the utilization phase, a servicePackage managed object may exist in the following states:

1) **Committed.** All pertinent Complex internal resources are allocated to the appropriate managed objects.

2) **Executing.** At least one transfer or storage service requested in the servicePackage is provided.

3) **Terminating.** After a *terminate* action invocation, the servicePackage managed object is waiting for the contained transfer and/or storage service instances that are active to enter state *done*.

4) **Done.** The Complex resources are released and any telecommunication association with another Complex (or with the spacecraft) is released.

5.17.5 ATTRIBUTES—ServicePackage MO CLASS

5.17.5.1 General

This managed object class has the following specific attributes:

a) **service-package-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (see annex C).

1) Data type/values: Character string.

2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **expiration-time**. The time the servicePackage lifetime ends. SLE Complex Management computes it after a successful validation. The *expiration-time* is the absolute time for which the last service provision or storage activity of a contained managed object is scheduled to terminate.

1) Data type/values: CCSDS time code.

2) Modifications:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

c) **service-package-state**. The state of the servicePackage processing within the Complex.

   1) Data type/values: See 5.17.4.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

d) **validation-result**. A summary of the result of the validation process. The validation process sets it.

   1) Data type/values:

      i) 'no result available' is the initial value;

      ii) 'passed' is set if the validation process is complete and no defect was discovered;

      iii) 'defects detected' is set if the validation process is complete and one or more defects were detected.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

e) **defect-records**. A set of records documenting defects discovered during validation. As defects are detected, the various records are added to the sequence.

   1) Data type/values: Set of records as specified in table 5-13. Initial value is 'empty'.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

f) **last-validation-request**. The time of the last validation request.

   1) Data type/values: CCSDS time code. Initial value shall be 'null'.

   2) Modifications:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

g) **remaining-time**. The remaining time (in hh:mm:ss <u>or</u> seconds) until the service package expires.

1) Data type/values: Unsigned integer. Initial value shall be the difference between the current time and *expiration-time*.

2) Modifications:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

h) **entered-committed**. The time when the servicePackage managed object entered the state *committed*.

1) Data type/values: CCSDS time code. Initial value shall be 'null'.

2) Modifications:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

i) **done-time**. The time when the managed object entered the state *done.*

1) Data type/values: CCSDS time code.

2) Modifications:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

Refer to table 5-16 for the servicePackage managed object attribute value summary.

Table 5-16: Attribute Value Summary—ServicePackage Managed Object

Attribute	Inherited From	Initial Value	Specified In
service-package-mo-id	—	by CM	CREATE.inv
expiration-time	—	by UM	CREATE.inv
service-package-state	—	= waiting	No
validation-result	—	= no result available	No
defect-records	—	= empty	No
last-validation-request	—	by CM	as needed
remaining-time	—	by CM	as needed
entered-committed	—	by CM	as needed
done-time	—	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

5.17.5.2 The Defect-records Attribute

Refer to table 5-17 for defect-records.

Table 5-17: Defect-Records

Record name	Purpose	Contents
Completeness defect	Indicates that a managed object required for successful validation was unavailable	– record name (data type character string) – unique record identifier (data type unsigned integer) – object identifier of the managed object type that was unavailable when the Service Package was validated
Conformance defect	Describes a non-conformance (detected during servicePackage validation) between an item in the servicePackage managed object (or its contained objects) with the applicable serviceAgreement managed object	– record name (data type character string) – unique record identifier (data type unsigned integer) – RDN of the nonconformant managed object – RDN of the managed object to which conformance is prescribed
Consistency defect	Describes an inconsistency (detected during servicePackage validation) between items in the servicePackage (attributes or objects contained)	– record name (data type character string) – unique record identifier (data type unsigned integer) – description of the inconsistency (data type character string) – list of the RDNs of the managed objects found to be inconsistent
Resource limitation defect	Indicates that Complex resources cannot support a transfer service request	– record name (data type character string) – unique record identifier (data type unsigned integer) – the kind of resource causing the problem (data type character string) – RDN of the managed object representing the transfer service request concerned

Record name	Purpose	Contents
Conflict defect	Anonymous conflict: Indicates one or more conflicts with one or more of the Complex's anonymous customer missions (i.e., conflicts with customer missions that have the *anonymity-flag* in the serviceAgreement managed object set to 'anonymity required').	– record name (data type character string) – unique record identifier(data type unsigned integer) – ultimate due time for resolution (see note 1) (undefined after creation, data type CCSDS time code). Ultimate due time is derived from • the earliest due time for service provision found in the servicePackages involved in the conflict • a comfortable margin reflecting the typical lead time for Complex internal configurations
	Conflict record: Describes one or more conflicts with ordinary customer missions (i.e., conflicts with customer missions that have the *anonymity-flag* in the serviceAgreement managed object set to 'anonymity not required').	– record name (data type character string) – unique record identifier (data type unsigned integer) – description of the conflict (data type character string) – RDN of the managed object causing the conflict – list of the DNs of the managed objects affected by the conflict (see notes 2 and 3) – ultimate due time for resolution: 'undefined' after servicePackage creation (see above).

NOTES

1 The ultimate due time indicates until when the definitive conflict resolution must be agreed upon and in effect between SLE Utilization Managements. If this cannot be achieved, the servicePackage, which is in the state under validation, cannot transition to executing.

2 A servicePackage may conflict with several other servicePackages, each of which may be involved in other conflicts as well. Iterative attempts may be required to resolve their conflicts. The manner in which this conflict is resolved is outside the scope of this draft Recommendation. It is part of the customer policy of each Complex.

3 The DN allows identification of the mission and spacecraft related to the affected managed objects.

4 In the case of anonymous conflicts, it is SLE Complex Management's responsibility to apply a suitable conflict resolution policy. For instance, SLE Complex Management may represent the anonymous missions in negotiations with ordinary missions.

During the validation process, the description of a newly detected defect is added to the set of *defect-records* as follows:

a) If non-conformance of the servicePackage with the serviceAgreement managed object is detected, a ‘conformance defect’ record is added.

b) If an inconsistency between managed objects contained in the same servicePackage managed object is detected, a ‘consistency defect’ record is added.

c) If missing managed objects are detected, a ‘completeness defect’ record is added.

d) If a conflict with another servicePackage managed object is detected, a ‘conflict’ record is added.

5.17.6 ACTIONS—ServicePackage MO CLASS

5.17.6.1 Validate Action

a) The **validate** action shall be confirmed if the *validate* action invocation is forwarded to the validating resource.

NOTE 1 – Depending on SLE Complex Management’s implementation, the validation activity may take considerable time or require human intervention. Therefore the validation process is a two-stage process: the first stage consists of the request from SLE Utilization Management to validate and the immediate confirmation to this request, the second stage is the actual validation and the reporting of its result.

b) This action has a **validation-mode** *action-parameter*. Possible values are as follows:

1) ‘commit’. SLE Utilization Management wants the service provision to take place and that the required Complex resources are reserved in due time.

2) ‘test’. SLE Utilization Management wants to check the feasibility of the servicePackage without reservation of Complex’s resources.

NOTE 2 – The fact that a servicePackage has successfully passed a validation in ‘test’ mode does not guarantee that a subsequent validation in ‘commit’ mode will succeed.

c) SLE Complex Management shall confirm the *validate* action and the servicePackage managed object shall transition to the state *under validation* if all of the following conditions are met:

1) if the *validation-mode* is ‘commit’, the state of the containing serviceAgreement managed object must be *applicable;*

NOTE 3 – If the *validation-mode* is ‘test’, the state of the serviceAgreement managed object is irrelevant.

2) the servicePackage managed object contains at least a ‘service used’ (xxx-ts-u) or a ‘service stored’ (xxx-st) managed object;

3) the servicePackage managed object is currently in the state *waiting*.

5.17.6.2 Terminate Action

a) After a **terminate** action invocation, no new Complex-internal activity relating to the servicePackage managed object or its contained managed objects shall be started.

   1) Contained managed objects that are already active shall remain active.

   2) The *expiration-time* shall be re-computed on the basis of the remaining activations of managed objects. The attribute shall be updated accordingly.

b) No specific *action-parameters* are associated with this action.

5.17.6.3 Abort Action

a) After an **abort** action invocation, any Complex-internal activity relating to the servicePackage managed object or its contained managed objects shall be aborted immediately. Upon receipt of the action invocation, the performer shall return an *aborted* state transition notification.

b) No specific *action-parameters* are associated with this action.

5.17.6.4 Decommit Action

a) A **decommit** action shall set the servicePackage managed object and all its contained managed objects to the state *waiting*, thus enabling editing of the servicePackage managed object itself and its contained managed objects.

b) No specific *action-parameters* are associated with this action.

c) A *decommit* action may only be invoked after the servicePackage managed object has successfully passed validation and therefore is in the *committed* state.

5.17.7 STATE-RELATED BEHAVIOR—ServicePackage MO CLASS

See figure 5-7 for the servicePackage state transition diagram. The first two states of the definition phase for managed objects contained in a servicePackage managed object are *waiting* and *under validation*. The exception is the NotificationLog managed object class, which has no states.

NOTES

1 GET operation requests are accepted in all states.

2 No general rules are defined in this subsection for the acceptance or rejection of SET operation requests. They are state- and attribute-dependent and hence are part of the attributes description.

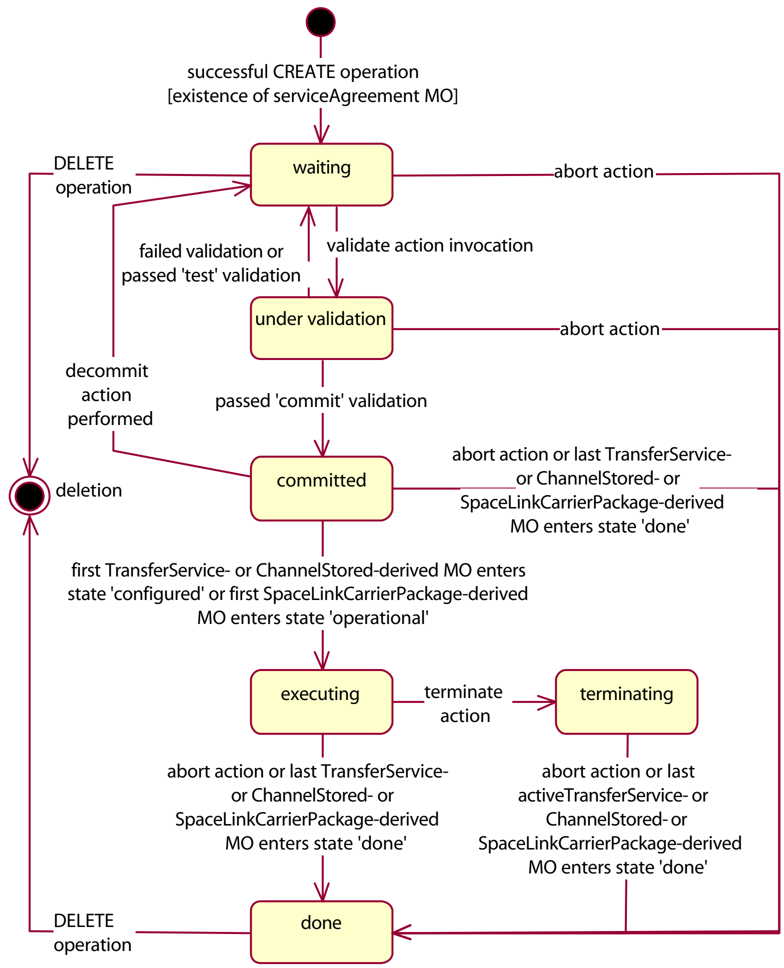


Figure 5-7: ServicePackage State Transition Diagram

5.17.7.1 WAITING State

5.17.7.1.1 Conditions for the servicePackage managed object entering the state *waiting*

a) At creation time.

b) If a validation failed.

c) If a ‘test’ servicePackage passed validation.

d) In response to a *decommit* action.

5.17.7.1.2 Actions and operations accepted and rejected by the servicePackage managed object in the state *waiting*

a) Accepts a *validate* action invocation.

b) Rejects a *terminate* action invocation.

c) Accepts an *abort* action invocation.

d) Rejects a *decommit* action invocation.

e) Accepts a DELETE invocation.

f) Accepts a CREATE invocation for contained managed objects.

5.17.7.2 UNDER VALIDATION State

When in the state *under validation*, the servicePackage managed object is subject to SLE Complex Management's validation process.

5.17.7.2.1 Conditions for the servicePackage managed object entering the state *under validation*

When a *validate* action is confirmed by SLE Complex Management and the validation process is started.

5.17.7.2.2 Actions and operations accepted and rejected by the servicePackage managed object in the state *under validation*

a) Rejects a *validate* action invocation.

b) Rejects a *terminate* action invocation.

c) Accepts an *abort* action invocation.

d) Rejects a *decommit* action invocation.

e) Rejects a DELETE invocation.

f) Rejects a CREATE invocation for contained managed objects.

5.17.7.3 COMMITTED State

5.17.7.3.1 Conditions for the servicePackage managed object entering the state *committed*

If validation is successfully performed and the qualifier 'commit' has been set.

5.17.7.3.2 Actions and operations accepted and rejected by the servicePackage managed object in the state *committed*

a) Rejects a *validate* action invocation.

b) Rejects a *terminate* action invocation.

c) Accepts an *abort* action invocation.

d) Accepts a *decommit* action invocation.

e) Rejects a DELETE invocation.

f) Rejects a CREATE invocation for contained managed objects.

5.17.7.4 EXECUTING State

5.17.7.4.1 Conditions for the servicePackage managed object entering the state *executing*

If it is already *committed* and the first contained TransferService- or ChannelStored-derived managed object enters state *configured,* or the first contained SpaceLinkCarrierPackage-derived managed object enters state *operational*.

5.17.7.4.2 Actions and operations accepted and rejected by the servicePackage managed object in the state *executing*

a) Rejects a *validate* action invocation.

b) Accepts a *terminate* action invocation.

c) Accepts an *abort* action invocation.

d) Rejects a *decommit* action invocation.

e) Rejects a DELETE invocation.

f) Rejects a CREATE invocation for contained managed objects.

5.17.7.5 TERMINATING State

When in the state *terminating*, the servicePackage managed object:

a) waits until all active contained managed objects enter state *done*;

b) causes contained managed objects that are not active to transition immediately to state *done* (the states in which managed objects are considered 'active' are listed in 4.10).

5.17.7.5.1 Conditions for the servicePackage managed object entering the state *terminating*

In response to a *terminate* action.

5.17.7.5.2 Actions and operations accepted and rejected by the servicePackage managed object in the state *terminating*:

a) Rejects a *validate* action invocation.

b) Rejects a *terminate* action invocation.

c) Accepts an *abort* action invocation.

d) Rejects a *decommit* action invocation.

e) Rejects a DELETE invocation.

f) Rejects a CREATE invocation for contained managed objects.

5.17.7.6 DONE State

5.17.7.6.1 Conditions for the servicePackage managed object entering the state *done*

a) If all TransferService-, ChannelStored- and SpaceLinkCarrierPackage-derived instances contained in the servicePackage have transitioned to state *done*.

b) If after a *terminate* action all contained managed objects have transitioned to state *done*.

c) If the servicePackage managed object is aborted by an *abort* action.

5.17.7.6.2 Actions and operations accepted and rejected by the servicePackage managed object in the state *done*

a) Rejects a *validate* action invocation.

b) Rejects a *terminate* action invocation.

c) Rejects an *abort* action invocation.

d) Rejects a *decommit* action invocation.

e) Accepts a DELETE invocation.

f) Rejects a CREATE invocation for contained managed objects.

5.17.8 NOTIFICATIONS—ServicePackage MO CLASS

A servicePackage managed object issues the following state transition notifications:

a) **Validation passed 'test'.** The managed object has transitioned from the state *under validation* to *waiting* after a successful test validation.

b) **Validation passed 'commit'.** The managed object has transitioned from the state *under validation* to *committed*.

c) **Validation failed.** The managed object has transitioned from the state *under validation* to *waiting* after an unsuccessful validation. A *validation failed* notification contains the additional information:

   Defect-records. A set of records that documents defects discovered during validation. See 5.16.5. Each record is specified in table A-2.

d) **Execution started.** The managed object has transitioned from the state *committed* to *executing* with the start of the first service instance.

e) **Execution completed.** The managed object has transitioned from the state *executing* to *done* with the completion of the last service instance.

f) **Terminated.** The managed object has transitioned to the state *done* due to a *terminate* action.

g) **Aborted.** The managed object has transitioned from the state *executing* to *done* with an *abort* action.

5.17.9 VALIDATION ASPECTS—ServicePackage MO CLASS

a) During performance of a *validate* action, SLE Complex Management checks the servicePackage for the following:

   1) conformance with the serviceAgreement;

   2) completeness and internal consistency;

   3) conflicts with already *committed* or *executing* servicePackage managed objects existing at the same Complex that are contained by one of the following:

      i) the same serviceAgreement;

      ii) another serviceAgreement managed object of the same mission;

      iii) a serviceAgreement of other missions.

   NOTE – This draft Recommendation does not specify the way this checking is performed. However, the results of the validation process must be reported.

b) The list of defects and the contents of their respective records are specified in 5.16.5.

c) If a managed object that is mandatory for a successful validation does not exist, the servicePackage managed object enters the state *waiting* and a ‘completeness defect’ shall be added to the servicePackage managed object’s set of *defect-records*.

d) If a non-conformance of the Service Package with the Service Agreement that contains it is detected, the servicePackage managed object enters the state *waiting,* and a ‘conformance defect’ shall be added to the servicePackage managed object’s set of *defect-records*.

e) If an inconsistency between managed objects contained in the same servicePackage managed object is detected, the servicePackage managed object enters the state *waiting,* and an ‘consistency defect’ shall be added to the servicePackage managed object’s set of *defect-records*.

f) If a conflict with another Service Package is detected, the servicePackage managed object enters the state *waiting,* and a ‘conflict defect’ shall be added to the servicePackage managed object’s set of *defect-records*.

NOTE – A conflict may also arise from limited availability of the Complex's resources. It is beyond the scope of this draft Recommendation to address how resource conflicts are resolved by the Complex.

g) If all checks have been passed successfully, the *validation-result* is set to 'passed'.

h) If all checks have been passed successfully, and the validation has been requested with the *validation-mode* 'commit', the servicePackage enters the state *committed*.

i) If all checks are passed but the validation was requested with the *validation-mode* 'test', the servicePackage enters the state *waiting*.

j) Independent of the validation qualifier, after a successful validation, the *expiration-time* is computed and the attribute updated accordingly.

k) If the service package contains either an f-tc-session-prod or f-tc-vc-prod managed object and does not contain an instance of R-cf-prod, a 'completeness defect' shall be added to the servicePackage managed object's set of *defect-records*.

5.18 SLEManagementEntity MANAGED OBJECT CLASS

5.18.1 PURPOSE

The SLEManagementEntity managed object class serves as the root of the SLE managed object inheritance tree. All SLE managed object classes are derived from it and inherit its properties.

5.18.2 INHERITANCE—SLEManagementEntity MO CLASS

This managed object class is the root of the inheritance tree.

5.18.3 OBJECTS CONTAINED—SLEManagementEntity MO CLASS

This managed object class is an abstract class; thus it cannot be instantiated and containment relationships do not apply.

5.18.4 STATES—SLEManagementEntity MO CLASS

No specific states are defined for this managed object class.

5.18.5 ATTRIBUTES—SLEManagementEntity MO CLASS

This managed object class has the following attribute: **read-me**. The text used as annotation to the human user of management information.

1) Data type/values: Character string.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

Refer to table 5-18 for the SLEmanagementEntity managed object attribute value summary.

Table 5-18: Attribute Value Summary—SLEManagementEntity Managed Object

Attribute	Inherited From	Initial Value	Specified In
read-me	—	by UM	CREATE.inv

5.18.6 ACTIONS—SLEManagementEntity MO CLASS

No specific actions are defined for this managed object class.

5.18.7 STATE-RELATED BEHAVIOR—SLEManagementEntity MO CLASS

No specific state-related behavior is defined for this managed object class.

5.18.8 NOTIFICATIONS—SLEManagementEntity MO CLASS

An SLEManagementEntity managed object issues the following notifications:

a) **Action success.** Notification issued on the successful completion of an action that could not be successfully completed at the time the action *result* was returned (see 4.8). The *action success* notification contains the following additional information:

   1) *Action-id* from the original ACTION invocation.

   2) *Invoke-id* from the original ACTION invocation.

b) **Action failure.** Notification issued on the ultimate failure of an action that could only be acknowledged at the time the action *result* was returned (see 4.8). The *action failure* notification contains the following additional information:

   1) *Action-id* from the original ACTION invocation.

   2) *Invoke-id* from the original ACTION invocation.

   3) *Diagnostic*. The reason for failure.

   NOTE – Although all SLE managed objects inherit these notifications, they are invoked only by those managed objects that manage resources that perform actions that may be deferred. The need for and ability to defer the performance of an action may depend on the implementation of the SLE Complex, and may differ for the various ACTION operations performed.

5.18.9 VALIDATION ASPECTS—SLEManagementEntity MO CLASS

No specific validation aspects are defined for this managed object class.

5.19 TelecommunicationInformationBase MANAGED OBJECT CLASS

5.19.1 PURPOSE

A telecommunicationInformationBase (TIB) managed object is a container object for technology-specific binding information.

5.19.2 INHERITANCE—TelecommunicationInformationBase MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

5.19.3 OBJECTS CONTAINED—TelecommunicationInformationBase MO CLASS

a) For each *responder-port-identifier* in the containing serviceAgreement managed object, the TIB shall contain a managed object that collects the technology-specific information required to support binding at the designated port. The classes that may be contained in the TIB are defined in *Mapping Rules,* reference [9].

b) In those cases where SLE Complex Management specifies the contained managed objects, SLE Complex Management has the option of changing the specification while the servicePackage is in the state *waiting*.

NOTE – Because no notifications are sent, it is vital that SLE Utilization Management checks the values of the identifiers after validation is complete, and not merely use the identifiers that were returned as part of the original creation of the managed objects.

5.19.4 STATES—TelecommunicationInformationBase MO CLASS

A TIB managed object has no states.

5.19.5 ATTRIBUTES—TelecommunicationInformationBase MO CLASS

This managed object class shall have the following specific attribute: **tib-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (see annex C).

1) Data type/values: Character string.

2) Modifications: This attribute shall not be modifiable using the SET operation.

Refer to table 5-19 for the telecommunicationInformationBase managed object attribute value summary.

Table 5-19: Attribute Value Summary—TelecommunicationInformationBase Managed Object

Attribute	Inherited From	Initial Value	Specified In
tib-mo-id	—	by CM	CREATE.inv
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

5.19.6 ACTIONS—TelecommunicationInformationBase MO CLASS

A TIB managed object performs no actions.

5.19.7 STATE-RELATED BEHAVIOR—TelecommunicationInformationBase MO CLASS

A TIB managed object has no states.

5.19.8 NOTIFICATIONS—TelecommunicationInformationBase MO CLASS

A TIB managed object issues no notifications.

5.19.9 VALIDATION ASPECTS—TelecommunicationInformationBase MO CLASS

No specific validation aspects are identified for this managed object.

5.20 TransferService MANAGED OBJECT CLASS

5.20.1 PURPOSE

5.20.1.1 General

This class collects the following common features of all SLE transfer services, whether they are used or provided:

a) states and state transitions behavior;

b) service access and telecommunication establishment information;

c) transmission-mode information.

5.20.1.2 Perception by SLE Utilization Management

a) In the definition phase, this object specifies a transfer service that the Complex uses from another Complex (or the Space Link), or provides to a mission end user (or another Complex).

b) In the utilization phase, the transferService managed object allows monitoring of the transfer service provision.

c) SLE Utilization Management shall be responsible for ensuring that the service access details and telecommunication establishment information match those of a mission user entity or of another SLE complex.

5.20.1.3 Perception by SLE Complex Management

a) SLE Complex Management shall be responsible for ensuring that adequate port resources and the necessary telecommunication capabilities are available during the utilization phase.

b) During performance of the initial BIND operation of a transfer service, security checks shall be performed based on the information in the securityInformationBase managed object.

5.20.1.4 Deletion Aspects

Subsection 5.20.7 contains information about the states in which a transferService managed object can be deleted.

5.20.2 INHERITANCE—TransferService MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

5.20.3 OBJECTS CONTAINED—TransferService MO CLASS

This managed object class is an abstract class; thus it cannot be instantiated and containment relationships do not apply.

5.20.4 STATES—TransferService MO CLASS

a) In the definition phase, a transferService managed object shall exist in one of the following states:

   1) **Waiting**. The managed object is created and may be edited or submitted to the validation process.

   2) **Under validation**. The servicePackage managed object is undergoing validation. The transferService managed object is static until validation is complete.

   3) **Validated**. The managed object is confirmed by the validation of the servicePackage managed object that contains it. All pertinent Complex-internal resources are allocated to the managed object.

b) In the utilization phase, a transferService managed object shall exist in one of the following states:

1) **Configured**. The provision period has started. An association between the managed object and the resource has been established and the resource has been successfully configured according to the managed object's attribute values.

2) **Ready**. A BIND operation of the transfer service has been successfully executed.

NOTE – This corresponds to the *active* state described in the SLE Transfer Service Specifications.

3) **Active**. A START operation of the transfer service has been successfully executed.

NOTE – This corresponds to the *active* state described in reference.

4) **Unavailable.** Service provision is not possible for Complex-internal reasons or that communication with another Complex is not possible.

5) **Completed**. The provision period has ended. Complex resources have been released.

5) **Done**. All contained managed objects, if any, have transitioned to *done*.

5.20.5 ATTRIBUTES—TransferService MO CLASS

This managed object class has the following specific attributes:

a) **current-ts-version.** The supported SLE transfer service version number.

   1) Data type/values: Unsigned integer.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

b) **lower-bound-reporting-period.** The minimum allowable value (in seconds) for the *reporting-cycle* parameter in the transfer service SCHEDULE-STATUS-REPORT operation.

   1) Data type/values: Unsigned integer.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **time-out-period.** The amount of time (in seconds) the invoker will wait for a response. If the return response to an operation invocation that needs confirmation does not arrive within this period, the invoker shall release the association by invoking the transfer service PEER-ABORT operation.

   1) Data type/values: Unsigned integer.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

d) **transfer-service-state**. The current state of the local SLE transfer service entity (i.e., the resource at the SLE Complex that supports the port-function for a transfer service instance).

   1) Data type/values: See 5.20.4. The initial value is ‘waiting’.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

e) **association-initiation-responsibility.** This attribute assigns the right to initiate a transfer service association to one of the SLE transfer service entities involved. SLE Utilization Management specifies the value at managed object creation time. SLE Utilization Management can change the value as long as the managed object is in the state *waiting*.

   1) Data type/values:

      i) ‘local’. The SLE transfer service entity at the SLE Complex initiates the SLE service association by invoking a transfer-service-BIND operation; the remote SLE transfer service entity responds to the BIND.

      ii) ‘remote’. The SLE transfer service entity at the remote site (another Complex or a Mission End User) initiates the SLE service association by invoking a transfer-service-BIND operation; the local SLE transfer service entity responds to the BIND.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

f) **responder-port-identifier.** The logical name of the responding SLE transfer service entity. *Responder-port-identifier* serves as a key into the TIB, which contains all technology-specific detail relevant for the establishment and operation of telecommunication between entities.

   1) Data type/value: Character string that must be unique within the scope of the TIB.

   2) Modifications: The original entity may change the name when the managed object is in the state *waiting*.

      i) If the original entity is SLE Utilization Management, SET shall be enabled when the managed object is in the state *waiting*.

      ii) If the original entity is SLE Complex Management, the method by which the name may be changed is outside the scope of this draft Recommendation.

   3) Dependencies: This attribute depends on the value of *association-initiation-responsibility*.

      i) If the local SLE service entity shall respond to a transfer-service-BIND (i.e., *association-initiation-responsibility* = ‘remote’), the name must be specified at creation time by the local SLE Complex Management.

ii) If the remote SLE service entity shall respond to a transfer-service-BIND (i.e., *association-initiation-responsibility* = 'local'), the name must be specified at creation time by SLE Utilization Management.

g) **local-sle-entity-identifier.** The logical name of the entity at the SLE Complex that implements the port function for the give transfer service. It is required as a key into the Security Information Base.

1) Data type/values: Character string that must be unique within the scope of the containing serviceAgreement managed object.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

h) **peer-sle-entity-access-list.** A set of *peer-entity-identifiers*, each of which identifies a peer service entity that is privileged to establish an SLE service association with the SLE entity represented by this transfer service managed object. Only one peer-sle-entity can bind with the local entity, and once that peer entity has bound, it is the only entity that can bind for the service instance provision period.

1) Data type/values: IA5String.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

i) **scheduled-period-start-time.** The start of the service instance provision period for the transfer service instance. Before this time, any transfer service BIND initiation fails. The value for this attribute is specified in the CREATE invocation.

1) Data type/values: CCSDS time code.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

j) **scheduled-period-stop-time.** The end of the service instance provision period for the transfer service instance. After this time, any transfer service BIND initiation fails. The value for this attribute is specified in the CREATE invocation.

1) Data type/values: CCSDS time code.

2) Modifications: SET shall be enabled during the definition phase.

k) **first-data-unit-time**. The time of transfer of the first data unit for the service instance.

1) Data type/values: CCSDS time code. Initial value shall be 'null'.

2) Modifications:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

l) **last-data-unit-time**. The time of transfer of the last data unit for the service instance.

1) Data type/values: CCSDS time code. Initial value shall be ‘null’.

2) Modifications:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

m) **service-instance-start**. The time the first service instance started.

1) Data type/values: CCSDS time code. Initial value shall be ‘null’.

2) Modifications:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

n) **service-instance-stop**. The time the last service instance ended.

1) Data type/values: CCSDS time code. Initial value shall be ‘null’.

2) Modifications:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

o) **provision-time**. The accumulated time (in seconds) the managed object was in the state *committed* until reporting time.

1) Data type/values: Unsigned integer. Initial value shall be ‘0’.

2) Modifications:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

p) **failed-binds**. The number of failed transfer service BIND operation invocations until reporting time.

1) Data type/values: Unsigned integer. Initial value shall be ‘0’.

2) Modifications:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

q) **unavailable-time**. The accumulated time (in seconds) the managed object was in the state *unavailable* until reporting time.

   1) Data type/values: Unsigned integer. Initial value shall be '0'.

   2) Modifications:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

r) **done-time**. The time when the managed object entered the state *done.*

   1) Data type/values: CCSDS time code. Initial value shall be 'null'.

   2) Modifications:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

Refer to table 5-20 for the transferService managed object attribute value summary.

Table 5-20: Attribute Value Summary—TransferService Managed Object

Attribute	Inherited From	Initial Value	Specified In
current-ts-version	—	by UM	CREATE.inv
lower-bound-reporting-period	—	by UM	CREATE.inv
time-out-period	—	by UM	CREATE.inv
transfer-service-state	—	= *waiting*	—
association-initiation-responsibility	—	by UM	CREATE.inv
responder-port-identifier	—	by UM	CREATE.inv
local-sle-entity-identifier	—	by CM	—
peer-sle-entity-access-list	—	by UM	CREATE.inv
scheduled-period-start-time	—	by UM	CREATE.inv
scheduled-period-stop-time	—	by UM	CREATE.inv
first-data-unit-time	—	by CM	when first data unit is transferred
last-data-unit-time	—	by CM	when last data unit is transferred
service-instance-start	—	= 'null'	when first service instance starts
service-instance-stop	—	= 'null'	when last service instance stops
provision-time	—	= 0	as needed
failed-binds	—	= 0	as needed
unavailable-time	—	= 0	as needed
done-time	—	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

5.20.6 ACTIONS—TransferService MO CLASS

a) An **abort** action shall stop production even after a successful validation.

b) No specific *action-parameters* are associated with this action.

5.20.7 STATE-RELATED BEHAVIOR—TransferService MO CLASS

5.20.7.1 General

See figure 5-8 for the transferService state transition diagram. Since transferService managed objects are ultimately contained by servicePackage managed objects, the first two states of the definition phase, *waiting* and *under validation*, are the same as for the ServicePackage managed object class. Refer to 5.17.4 and 5.17.7 for a discussion of definition phase states and state transitions associated with the servicePackage managed object and its contained objects. We begin our discussion with the *validated* state.

NOTES

1 GET operation requests are accepted in all states.

2 No general rules are defined in this subsection for the acceptance or rejection of SET operation requests. They are state- and attribute-dependent and hence are part of the attributes description.

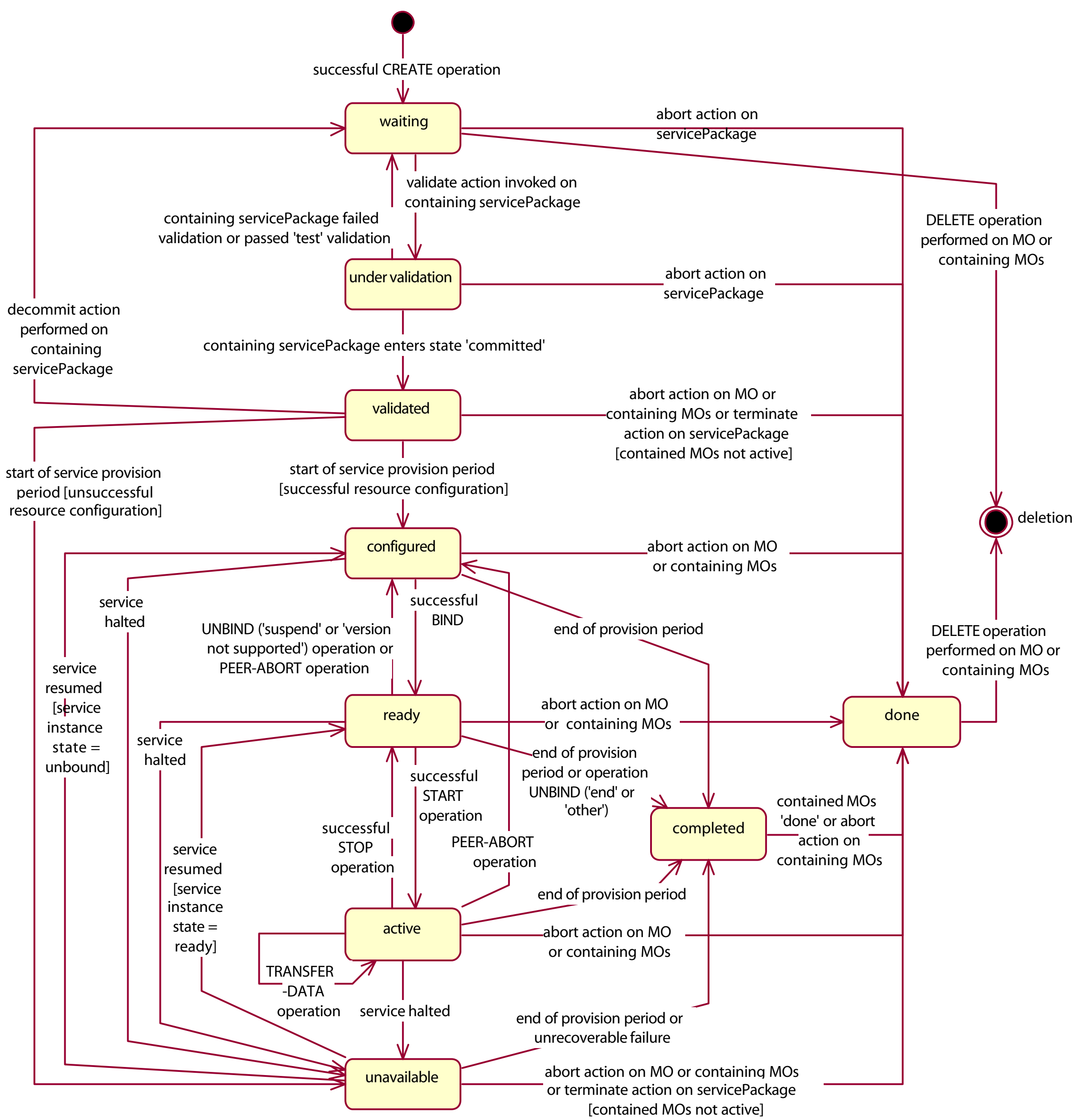


Figure 5-8: TransferService State Transition Diagram

5.20.7.2 VALIDATED State

5.20.7.2.1 Conditions for the transferService managed object entering the state validated

If it passed the validation process and the servicePackage managed object that contains it is *committed*.

5.20.7.2.2 Actions and operations accepted and rejected by the transferService managed object in the state *validated*

a) Rejects a DELETE invocation.

b) Rejects a CREATE invocation for contained managed objects.

c) Accepts an *abort* action invocation.

5.20.7.2.3 Actions and operations on containing managed objects that affect the transferService managed object when in the state *validated*

a) *Decommit* action performed on the containing servicePackage managed object causes the transferService managed object to transition to *waiting*.

b) *Terminate* action performed on the containing servicePackage managed object causes the transferService managed object to transition to *done* if none of its contained managed objects is active (the states in which managed objects are considered 'active' are listed in 4.10).

c) *Abort* action invoked on the containing managed object causes the transferService managed object to transition to *done*.

d) *Terminate* action performed on the containing channelStored managed object causes the transferService managed object to transition to *done*.

5.20.7.3 CONFIGURED State

5.20.7.3.1 General

When in the state *configured*, the transferService managed object:

a) enables the performance of a BIND operation by the respective transfer service resource;

b) restores attributes to values prior to the BIND operation, in case it has entered the state *configured* after an operation UNBIND ('suspend' or 'version not supported') performed when in the state *ready*.

5.20.7.3.2 Conditions for the transferService managed object entering the state *configured*

a) At the start of service instance provision period, if the SLE Complex Management has successfully configured the necessary resources for telecommunication with the other Complex and for transfer service processing.

b) If the respective transfer service resource has successfully performed the following:

   1) a transfer-service-UNBIND initiation with 'unbind-reason' set to 'suspend' or 'version not supported' when in the state *ready*;

   2) a transfer-service-PEER-ABORT with diagnostic parameter set to any of the possible values except to 'end of service instance provision period', when in either the state *ready* or *active*.

c) When service is resumed, when in the state *unavailable*, if the transfer service instance is in state 'unbound'.

5.20.7.3.3 Actions and operations accepted and rejected by the transferService managed object in the state *configured*:

a) Rejects a DELETE invocation.

b) Rejects a CREATE invocation for contained managed objects'.

c) Accepts an *abort* action invocation.

5.20.7.3.4 Actions and operations on containing managed objects that affect the transferService managed object when in the state *configured*

a) *Abort* action performed on the containing managed objects causes a state transition to *done*.

b) *Terminate* action performed on the containing channelStored managed object causes the transferService managed object to transition to *done*.

5.20.7.4 READY State

5.20.7.4.1 General

When in the state *ready*, the transferService managed object enables the transfer service resource to perform START transfer service operation.

5.20.7.4.2 Conditions for the transferService managed object entering the state *ready*

After a BIND operation has been successfully performed by the transfer service resource.

5.20.7.4.3 Actions and operations accepted and rejected by the transferService managed object in the state *ready*

a) Rejects a DELETE invocation.

b) Rejects a CREATE invocation for contained managed objects.

c) Accepts an *abort* action invocation.

5.20.7.4.4 Actions and operations on containing managed objects that affect the transferService managed object when in the state *ready*

a) *Abort* action performed on containing managed objects causes a state transition to *done*.

b) *Terminate* action performed on the containing channelStored managed object causes the transferService managed object to transition to *done*.

5.20.7.5 ACTIVE State

5.20.7.5.1 General

When in the state *active*, the transferService managed object enables the transfer service resource to perform all other transfer service operations.

5.20.7.5.2 Conditions for the transferService managed object entering the state *active*

When a START operation has been successfully performed by the transfer service resource.

5.20.7.5.3 Actions and operations accepted and rejected by the transferService managed object in the state *active*

a) Rejects a DELETE invocation.

b) Rejects a CREATE invocation for contained managed objects.

c) Accepts an *abort* action invocation.

5.20.7.5.4 Actions and operations on containing managed objects that affect the transferService managed object when in the state *active*

a) *Abort* action performed on the containing managed objects causes a state transition to *done*.

b) *Terminate* action performed on the containing channelStored managed object causes the transferService managed object to transition to *done*.

5.20.7.6 UNAVAILABLE State

5.20.7.6.1 General

When in the state *unavailable*, the transferService managed object waits for service to be resumed.

5.20.7.6.2 Conditions for the transferService managed object entering the state *unavailable*

If transfer service is halted because:

a) the respective transfer service resource is defective and is unable to perform the specific service;

b) at the start of the service instance provision period, the resources cannot be configured;

c) underlying telecommunication capabilities are corrupted.

5.20.7.6.3 Actions and operations accepted and rejected by the transferService managed object in the state *unavailable*

a) Rejects a DELETE invocation.

b) Rejects a CREATE invocation for contained managed objects.

c) Accepts an *abort* action invocation.

5.20.7.6.4 Actions and operations on containing managed objects that affect the transferService managed object when in the state *unavailable*

a) *Terminate* action performed on the containing servicePackage managed object causes a state transition to *done* if none of its contained managed objects is active (the states in which managed objects are considered ‘active’ are listed in 4.10).

b) *Abort* action invoked on the containing managed objects causes a state transition to *done*.

c) *Terminate* action performed on the containing channelStored managed object causes the transferService managed object to transition to state *done*.

5.20.7.7 COMPLETED State

5.20.7.7.1 General

When the state *completed* is entered, the transferService managed object is dissociated from its resources.

5.20.7.7.2 Conditions for the transferService managed object entering the state *completed*

a) If the transfer service instance provision period is over.

b) If recovery from corruption cannot be achieved.

c) If the supporting transfer service port received:

   1) a transfer-service-UNBIND initiation with 'unbind-reason' set to 'end' or 'other';

   2) a transfer-service-PEER-ABORT with diagnostic parameter set to 'end of service instance provision period'.

5.20.7.8 Actions and operations accepted and rejected by the transferService managed object in the state *completed*

a) Rejects a DELETE invocation.

b) Rejects a CREATE invocation for contained managed objects.

c) Rejects an *abort* action invocation.

5.20.7.9 Actions and operations on containing managed objects that affect the transferService managed object when in the state *completed*

a) *Abort* action performed on the containing managed objects causes the transferService managed object to transition to state *done*.

b) *Terminate* action performed on the containing channelStored managed object causes the transferService managed object to transition to state *done*.

5.20.7.10 Done State

5.20.7.10.1 Conditions for the transferService managed object entering the state *done*

a) If all of contained managed objects have entered state *done*, when managed object is in state *completed*.

b) If a *terminate* action is performed on the containing servicePackage managed object when the transferService managed object is in state *validated* or *unavailable,* and none of the contained managed objects is active (the states in which managed objects are considered 'active' are listed in 4.10).

c) When an *abort* action is performed on the managed object or on the containing managed objects.

5.20.7.10.2 Actions and operations accepted and rejected by the transferService managed object in the state *done*

a) Accepts a DELETE invocation.

b) Rejects a CREATE invocation for contained managed objects.

c) Rejects an *abort* action invocation.

5.20.7.10.3 Actions and operations on containing managed objects that affect the transferService managed object when in the state *done*

DELETE operation performed on the containing managed objects causes the transferService managed object to be deleted.

5.20.8 NOTIFICATIONS—TransferService MO CLASS

a) **Start of provision period.** The managed object has transitioned from the state *validated* to *configured*.

b) **End of provision period.** The managed object has transitioned to the state *completed,* because the service instance provision period ended.

c) **Service halted**. The managed object has transitioned from the state *configured* or *ready* or *active* to the state *unavailable*. This notification contains the additional information:

   Reason for interruption. An implementation-specific value within a specific Complex. The value is provided by SLE Complex Management. It is a character string.

d) **Service resumed**. The managed object has transitioned from the state *unavailable* to *configured* or *ready*.

e) **Unrecoverable failure.** The managed object has transitioned from the state *unavailable* to *completed*.

f) **Terminated.** The managed object has transitioned from the state *validated* or *unavailable* to *done* due to a *terminate* action on the containing servicePackage managed object.

5.20.9 VALIDATION ASPECTS—TransferService MO CLASS

No specific validation aspects are defined for this managed object class.

6 DATA CHANNEL TREE MANAGED OBJECT DEFINITIONS

6.1 DataChannelProfile MANAGED OBJECT CLASS

6.1.1 PURPOSE

An instance of the DataChannelProfile managed object class contains the channel tree elements for the Service Agreement.

6.1.2 INHERITANCE—DataChannelProfile MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.1.3 OBJECTS CONTAINED—DataChannelProfile MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero, one, or more instances of the F-tc-modulation-channel managed object class. This class is specified in 6.10.

b) Zero, one, or more instances of the F-cltu-channel managed object class. This class is specified in 6.5.

c) Zero, one, or more instances of the F-tc-frame-channel managed object class. This class is specified in 6.9.

d) Zero, one, or more instances of the F-af-channel managed object class. This class is specified in 6.2.

e) Zero, one, or more instances of the F-proto-vcdu-channel managed object class. This class is specified in 6.7.

f) Zero, one, or more instances of the R-af-channel managed object class. This class is specified in 6.15.

g) Zero, one, or more instances of the R-master-channel managed object class. This class is specified in 6.19.

6.1.4 STATES—DataChannelProfile MO CLASS

A DataChannelProfile managed object has no states.

6.1.5 ATTRIBUTES—DataChannelProfile MO CLASS

The DataChannelProfile managed object class shall have the following class-specific attribute:

a) **data-channel-profile-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/values: Character string.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

b) **Attribute Value Summary**. Refer to table 6-1.

Table 6-1: Attribute Value Summary—DataChannelProfile Managed Object

Attribute	Inherited From	Initial Value	Specified In
data-channel-profile-mo-id	—	—	—
read-me	SLEManagementEntity (5.18)	—	—

6.1.6 ACTIONS—DataChannelProfile MO CLASS

A DataChannelProfile managed object performs no actions.

6.1.7 STATE-RELATED BEHAVIOR—DataChannelProfile MO CLASS

A DataChannelProfile managed object performs no state transitions.

6.1.8 NOTIFICATIONS—DataChannelProfile MO CLASS

A DataChannelProfile managed object issues no notifications.

6.1.9 VALIDATION ASPECTS—DataChannelProfile MO CLASS

There are no specific validation aspects identified for this managed object class.

6.2 F-af-channel MANAGED OBJECT CLASS

NOTE – Although the *Cross Support Reference Model* (reference [1]) specifies a set of forward AOS SLE data channels, the current suite of SLE services do not contain any forward AOS services to carry these forward AOS SLE data channels. This managed object class definition subsection is a placeholder for one that will be needed when forward AOS services are specified.

6.2.1 PURPOSE

An instance of the F-af-channel managed object class represents the characteristics of a forward all frames data channel. All frames of the F-af-channel are AOS (Version 2) frames.

6.2.2 INHERITANCE—F-af-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.3 F-bitstream-channel MANAGED OBJECT CLASS

NOTE – Although the *Cross Support Reference Model* (reference [1]) specifies a set of forward AOS SLE data channels, the current suite of SLE services do not contain any forward AOS services to carry these forward AOS SLE data channels. This managed object class definition subsection is a placeholder for one that will be needed when forward AOS services are specified.

6.3.1 PURPOSE

An instance of the F-bitstream-channel managed object class represents the characteristics of a forward bitstream data channel.

6.3.2 INHERITANCE—F-bitstream-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.4 F-c/vcdu-channel MANAGED OBJECT CLASS

NOTE – Although the *Cross Support Reference Model* (reference [1]) specifies a set of forward AOS SLE data channels, the current suite of SLE services do not contain any forward AOS services to carry these forward AOS SLE data channels. This managed object class definition subsection is a placeholder for one that will be needed when forward AOS services are specified.

6.4.1 PURPOSE

An instance of the F-c/vcdu-channel managed object class represents the characteristics of a forward C/VCDU data channel. All frames of the F-c/vcdu-channel are AOS (Version 2) frames.

6.4.2 INHERITANCE—F-c/vcdu-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.5 F-cltu-channel MANAGED OBJECT CLASS

6.5.1 PURPOSE

An instance of the F-cltu-channel managed object class represents the characteristics of a Command Link Transmission Unit (CLTU) data channel.

6.5.2 INHERITANCE—F-cltu-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.5.3 OBJECTS CONTAINED—F-cltu-channel MO CLASS

An instance of this concrete managed object class shall contain zero or one instance of the F-tc-frame-channel managed object class. This class is specified in 6.9.

6.5.4 STATES—F-cltu-channel MO CLASS

A F-cltu-channel managed object has no states.

6.5.5 ATTRIBUTES—F-cltu-channel MO CLASS

The F-cltu-channel managed object class shall have the following class-specific attributes:

a) **cltu-channel-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

    1) Data type/value: Character string.

    2) Modifications: This attribute shall not be modifiable using the SET operation.

b) **tc-code-block-length**. The length of the Telecommand codeblock, in octets.

    1) Data type/value: 5, 6, 7, 8.

    2) Modifications: This attribute shall not be modifiable using the SET operation.

c) **max-length-cltu**. The maximum length (in octets) of the CLTUs that are carried by the CLTU data channel.

    1) Data type/value: Unsigned integer.

    2) Modifications: This attribute shall not be modifiable using the SET operation.

d) **cltu-randomization**. Whether the Channel Service Randomization Procedure (subsection 3.3.1 of *Telecommand Part 1—Channel Service,* reference [6]) is to be applied for this service package.

    1) Data type/values:

        i) 'always randomize'. Apply randomization for all service packages.

        ii) 'never randomize'. Never apply randomization for any service package.

        iii) 'package optional'. Application is specified on a per-service package basis.

    2) Modifications: This attribute shall not be modifiable using the SET operation.

3) Dependencies: This attribute must be specified in the f-cltu-prod managed object if the value of the *cltu-randomization-options* attribute in the corresponding f-cltu-channel managed object is 'package optional'. If the value of the corresponding *cltu-randomization-options* attribute is either 'always randomize' or 'never randomize', the value of the *cltu-randomization* attribute need not be specified. However, if it is specified, it must match the value of the corresponding *cltu-randomization-options* attribute or a validation error shall result.

e) **sle-data-channel-dn**. The distinguished name of the data channel.

1) Data type/values: DN.

2) Modifications: This attribute shall not be modifiable using the SET operation.

f) **protocol-abort-mode**. This attribute specifies whether transmission of buffered CLTUs shall be continued following a protocol abort of the forward SLE transfer service instance(s) supplying the input data for this CLTU data channel.

1) Data type/values:

i) 'flush'. All buffered CLTUs are to be discarded.

ii) 'continue'. All buffered CLTUs are to be transmitted.

2) Modifications: This attribute shall not be modifiable using the SET operation.

g) **Attribute Value Summary**. Refer to table 6-2.

Table 6-2: Attribute Value Summary—F-cltu-channel Managed Object

Attribute	Inherited From	Initial Value	Specified In
cltu-channel-mo-id	—	—	—
tc-code-block-length	—	—	—
max-length-cltu	—	—	—
cltu-randomization	—	—	—
sle-data-channel-dn	—	—	—
protocol-abort-mode	—	—	—
read-me	SLEManagementEntity (5.18)	—	—

NOTE – In accordance with the naming conventions specified for SLE data channel names in annex C, the DN for the F-cltu-channel has the component RDNs (mission-SC, CLTU).

6.5.6 ACTIONS—F-cltu-channel MO CLASS

A F-cltu-channel managed object performs no actions.

6.5.7 STATE-RELATED BEHAVIOR—F-cltu-channel MO CLASS

A F-cltu-channel managed object performs no state transitions.

6.5.8 NOTIFICATIONS—F-cltu-channel MO CLASS

A F-cltu-channel managed object issues no notifications.

6.5.9 VALIDATION ASPECTS—F-cltu-channel MO CLASS

No specific validation aspects are identified for this managed object class.

6.6 F-insert-channel MANAGED OBJECT CLASS

NOTE – Although the *Cross Support Reference Model* (reference [1]) specifies a set of forward AOS SLE data channels, the current suite of SLE services do not contain any forward AOS services to carry these forward AOS SLE data channels. This managed object class definition subsection is a placeholder for one that will be needed when forward AOS services are specified.

6.6.1 PURPOSE

An instance of the F-insert-channel managed object class represents the characteristics of a forward insert data channel.

6.6.2 INHERITANCE—F-insert-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.7 F-proto-vcdu-channel MANAGED OBJECT CLASS

NOTE – Although the *Cross Support Reference Model* (reference [1]) specifies a set of forward AOS SLE data channels, the current suite of SLE services do not contain any forward AOS services to carry these forward AOS SLE data channels. This managed object class definition subsection is a placeholder for one that will be needed when forward AOS services are specified.

6.7.1 PURPOSE

An instance of the F-proto-vcdu-channel managed object class represents the characteristics of a forward protoVCDU data channel. All frames of the F-proto-vcdu-channel are AOS (Version 2) frames.

6.7.2 INHERITANCE—F-proto-vcdu-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.8 F-sp-channel MANAGED OBJECT CLASS

6.8.1 PURPOSE

An instance of the F-sp-channel managed object class represents the characteristics of a forward space packet data channel.

6.8.2 INHERITANCE—F-sp-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.8.3 OBJECTS CONTAINED—F-sp-channel MO CLASS

An instance of this concrete managed object class contains no other managed objects.

6.8.4 STATES—F-sp-channel MO CLASS

A F-sp-channel managed object has no states.

6.8.5 ATTRIBUTES—F-sp-channel MO CLASS

The F-sp-channel managed object class shall have the following class-specific attributes:

a) **f-sp-channel-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/values: Character string.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

b) **max-packet-length**. The upper bound on the length (in octets) of CCSDS packets on this space packet channel for the life of the service agreement.

   1) Data type/value: Integer in the range [6..65542], inclusive.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

c) **aggregation-options**. Whether packet aggregation will always be allowed, always disallowed, or allowed on a service package basis.

   1) Data type/ values:

      i) 'always-allowed' if aggregation of packets from this f-sp-channel is always to be allowed.

      ii) 'never-allowed' if aggregation of packets from this f-sp-channel is never to be allowed.

      iii) 'aggregation-optional' if aggregation is to be selectable on a service package basis. If the value is ‘aggregation-optional’, a value must be assigned to the 'aggregation-allowed' attribute of the corresponding f-sp-ts-p managed object contained in the servicePackage managed object.

2) Modifications: This attribute shall not be modifiable using the SET operation.

d) **sle-data-channel-dn**. The distinguished name of the data channel.

1) Data type/value: DN.

2) Modifications: This attribute shall not be modifiable using the SET operation.

e) **Attribute Value Summary**. Refer to table 6-3.

Table 6-3: Attribute Value Summary—F-sp-channel Managed Object

Attribute	Inherited From	Initial Value	Specified In
f-sp-channel-mo-id	—	—	—
max-packet-length	—	—	—
aggregation-options	—	—	—
sle-data-channel-dn	—	—	—
read-me	SLEManagementEntity (5.18)	—	—

NOTE – In accordance with the naming conventions specified for SLE data channel names in annex C, the DN for the F-sp-channel has the component RDNs (mission-SC, fwd-TC-Frame, fwd-TC-mux, fwd-space-pkt).

6.8.6 ACTIONS—F-sp-channel MO CLASS

A F-sp-channel managed object performs no actions.

6.8.7 STATE-RELATED BEHAVIOR—F-sp-channel MO CLASS

A F-sp-channel managed object performs no state transitions.

6.8.8 NOTIFICATIONS—F-sp-channel MO CLASS

A F-sp-channel managed object issues no notifications.

6.8.9 VALIDATION ASPECTS—F-sp-channel MO CLASS

There are no specific validation aspects identified for this managed object class.

6.9 F-tc-frame-channel MANAGED OBJECT CLASS

6.9.1 PURPOSE

An instance of the F-tc-frame-channel managed object class represents the characteristics of a TC frame data channel.

6.9.2 INHERITANCE—F-tc-frame-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.9.3 OBJECTS CONTAINED—F-tc-frame-channel MO CLASS

An instance of this concrete managed object class shall contain zero, one, or more instances of the F-tc-virtual-channel managed object class. This class is specified in 6.13.

NOTE – The F-tc-virtual-channels contained by a single F-tc-frame-channel managed object must all be affiliated with the same Spacecraft Identifier (SCID).

6.9.4 STATES—F-tc-frame-channel MO CLASS

An F-tc-frame-channel managed object has no states.

6.9.5 ATTRIBUTES—F-tc-frame-channel MO CLASS

The F-tc-frame-channel managed object class shall have the following class-specific attributes:

a) **tc-frame-channel-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/values: Character string.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

b) **max-frame-rate**. The maximum rate (in frames per second) that TC Frames will be carried by the data channel.

   1) Data type/values: Positive integer.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

c) **sle-data-channel-dn**. This attribute contains the distinguished name of the data channel.

   1) Data type/values: DN.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

d) **Attribute Value Summary**. Refer to table 6-4.

Table 6-4: Attribute Value Summary—F-tc-frame-channel Managed Object

Attribute	Inherited From	Initial Value	Specified In
tc-frame-channel-mo-id	—	—	—
max- frame-rate	—	—	—
sle-data-channel-dn	—	—	—
read-me	SLEManagementEntity (5.18)	—	—

NOTE – In accordance with the naming conventions specified for SLE data channel names in annex C, the DN for the F-tc-frame-channel has the component RDNs (mission-SC, fwd-TC-Frame).

6.9.6 ACTIONS—F-tc-frame-channel MO CLASS

An F-tc-frame-channel managed object performs no actions.

6.9.7 STATE-RELATED BEHAVIOR—F-tc-frame-channel MO CLASS

An F-tc-frame-channel managed object performs no state transitions.

6.9.8 NOTIFICATIONS—F-tc-frame-channel MO CLASS

An F-tc-frame-channel managed object issues no notifications.

6.9.9 VALIDATION ASPECTS—F-tc-frame-channel MO CLASS

a) If the F-tc-frame-channel managed object is contained by a F-cltu-channel managed object in the service agreement, the fwd_TC_Frame attribute of the DN of the TC frame data channel must have the same value as the cLTU attribute of the DN of the CLTU data channel.

b) If the F-tc-frame-channel managed object contains multiple F-tc-virtual-channel managed objects, the SCID component of the ret_CF attribute of the DNs of the contained TC virtual channels must all have the same value.

6.10 F-tc-modulation-channel MANAGED OBJECT CLASS

6.10.1 PURPOSE

An instance of the F-tc-modulation-channel managed object class represents the characteristics that are to be applied to the modulation of a CLTU data channel onto a forward symbol stream.

6.10.2 INHERITANCE—F-tc-modulation-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.10.3 OBJECTS CONTAINED—F-tc-modulation-channel MO CLASS

An instance of this concrete managed object class shall contain zero or one instance of the F-cltu-channel managed object class. The F-cltu-channel managed object class is specified in 6.5.

6.10.4 STATES—F-tc-modulation-channel MO CLASS

An f-tc-modulation-channel managed object has no states.

6.10.5 ATTRIBUTES—F-tc-modulation-channel MO CLASS

The f-tc-modulation-channel managed object class shall have the following class-specific attributes:

a) **f-tc-modulation-channel-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

    1) Data type/values: Character string.

    2) Modifications: This attribute shall not be modifiable using the SET operation.

b) **plop-in-effect.** This attribute is used to specify the Physical Layer Operations Procedure (PLOP) to be used on the modulation channel.

    1) Data type/values:

        i) plop-1.

        ii) plop-2.

    2) Modifications: This attribute shall not be modifiable using the SET operation.

c) **acquisition-sequence-length.** This attribute specifies the length (in bits) of the acquisition sequence used in Telecommand Carrier Modulation Mode (CMM) 2.

    1) Data type/values: Positive integer. The default value is 16.

    2) Modifications: This attribute shall not be modifiable using the SET operation.

d) **minimum-idle-length.** This attribute specifies the length (in bits) of the Idle Sequence to be used in Carrier Modulation Mode (CMM) 4 when PLOP-1 is used.

    1) Data type/values: positive integer in range [0..2040].

    2) Modifications: This attribute shall not be modifiable using the SET operation.

    3) Dependencies: Only used if *plop-in-effect* = plop-1, otherwise value = 0.

NOTE – This attribute is meaningful only for PLOP-1, which permits an optional modulation of the carrier with an idle sequence for the purposes of clocking bits through the PLOP. If the optional idle sequence is not used, the value of this attribute is zero.

e) **min-delay-time**. The minimum value (in microseconds) allowed for the *delay-time* parameter of the TRANSFER-DATA invocations carried by the transfer service instance.

   1) Data type/values: Unsigned integer.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

f) **Attribute Value Summary**. Refer to table 6-5.

Table 6-5: Attribute Value Summary—f-tc-modulation-channel Managed Object

Attribute	Inherited From	Initial Value	Specified In
f-tc-modulation-channel-mo-id	—	—	—
plop-in-effect	—	—	—
acquisition-sequence-length	—	—	—
minimum-idle-length	—	—	—
min-delay-time	—	—	—
read-me	SLEManagementEntity (5.18)	—	—

6.10.6 ACTIONS—F-tc-modulation-channel MO CLASS

An f-tc-modulation-channel managed object performs no actions.

6.10.7 STATE-RELATED BEHAVIOR—F-tc-modulation-channel MO CLASS

An f-tc-modulation-channel managed object performs no state transitions.

6.10.8 NOTIFICATIONS—F-tc-modulation-channel MO CLASS

An f-tc-modulation-channel managed object issues no notifications.

6.10.9 VALIDATION ASPECTS—F-tc-modulation-channel MO CLASS

No specific validation aspects are defined for this managed object class.

6.11 F-tc-multiplex-channel MANAGED OBJECT CLASS

6.11.1 PURPOSE

An instance of the F-tc-multiplex-channel managed object class represents the characteristics of a TC-Multiplex data channel. The TC-Multiplex data channel multiplexes TC segments or groups of Forward Space Packets destined for a single TC virtual channel.

6.11.2 INHERITANCE—F-tc-multiplex-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.11.3 OBJECTS CONTAINED—F-tc-multiplex-channel MO CLASS

An instance of this concrete managed object class shall contain one or more instances of the F-sp-channel managed object class. This class is specified in 6.8.

6.11.4 STATES—F-tc-multiplex-channel MO CLASS

An F-tc-multiplex-channel managed object has no states.

6.11.5 ATTRIBUTES—F-tc-multiplex-channel MO CLASS

The F-tc-multiplex-channel managed object class shall have the following class-specific attributes:

a) **tc-mux-channel-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/values: Character string.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

b) **segmentation-options.** Whether segmentation will be presence/absence in the TC-Multiplex data channel will be constant, or will change on a service package basis.

   1) Data type/values:

      i) 'segmentation-always-used'. Segmentation shall be used in all service packages.

      ii) 'segmentation-never-used'. Segmentation shall never be used in any service packages.

      iii) 'segmentation-optional'. Segmentation shall be optional on a service package basis.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

3) Dependencies: If the value is 'segmentation-optional', a value must be assigned to the *segmentation-flag* attribute of the corresponding f-vc-seg-prod managed object contained in the servicePackage managed object.

c) **map-multiplexing-scheme-options**. Whether the Multiplexer Access Point (MAP) multiplexing scheme for the TC-Multiplex data channel will be constant, or will change on a service package basis.

   1) Data type/values:

      i) 'always-fifo'. The requests shall be served first-in, first-out in all service packages.

      ii) 'always-absolute-priority'. The highest priority request shall always be served first in all service packages.

      iii) 'always-polling-vector'. The requests shall always be served according to a predefined polling vector in all service packages.

      iv) 'multiplexing-scheme-optional'. The MAP multiplexing scheme shall be selectable on a service package basis.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

   3) Dependencies:

      i) This attribute shall only apply if *segmentation-options* has a value of 'segmentation-always-used' or 'segmentation-optional'.

      ii) If the value is 'multiplexing-scheme-optional', a value must be assigned to the *map-multiplexing-control* attribute of the corresponding f-vc-seg-prod managed object contained in the servicePackage managed object.

d) **sle-data-channel-dn.** The distinguished name of the data channel.

   1) Data type/values: DN.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

e) **Attribute Value Summary**. Refer to table 6-6.

Table 6-6: Attribute Value Summary—F-tc-multiplex-channel Managed Object

Attribute	Inherited From	Initial Value	Specified In
tc-mux-channel-mo-id	—	—	—
segmentation-options	—	—	—
map-multiplexing-scheme-options	—	—	—
sle-data-channel-dn	—	—	—
read-me	SLEManagementEntity (5.18)	—	—

NOTE – In accordance with the naming conventions specified for SLE data channel names in annex C, the DN for the F-tc-multiplex-channel has the component RDNs (mission-SC, fwd-TC-Frame, fwd-TC-mux).

6.11.6 ACTIONS—F-tc-multiplex-channel MO CLASS

A F-tc-multiplex-channel managed object performs no actions.

6.11.7 STATE-RELATED BEHAVIOR—F-tc-multiplex-channel MO CLASS

A F-tc-multiplex-channel managed object performs no state transitions.

6.11.8 NOTIFICATIONS—F-tc-multiplex-channel MO CLASS

A F-tc-multiplex-channel managed object issues no notifications.

6.11.9 VALIDATION ASPECTS—F-tc-multiplex-channel MO CLASS

There are no specific validation aspects identified for this managed object class.

6.12 F-tc-vca-channel MANAGED OBJECT CLASS

6.12.1 PURPOSE

An instance of the F-tc-vca-channel managed object class represents the characteristics of a TC-VCA data channel.

6.12.2 INHERITANCE—F-tc-vca-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.12.3 OBJECTS CONTAINED—F-tc-vca-channel MO CLASS

An instance of this concrete managed object class shall contain zero or one instance of the F-tc-multiplex-channel managed object class. This class is specified in 6.11.

6.12.4 STATES—F-tc-vca-channel MO CLASS

An F-tc-vca-channel managed object has no states.

6.12.5 ATTRIBUTES—F-tc-vca-channel MO CLASS

The F-tc-vca-channel managed object class shall have the following class-specific attributes:

a) **tc-vca-channel-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/values: Character string.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

b) **sle-data-channel-dn**. The distinguished name of the data channel.

   1) Data type/values: DN.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

c) **Attribute Value Summary**. Refer to table 6-7.

Table 6-7: Attribute Value Summary—F-tc-vca-channel Managed Object

Attribute	Inherited From	Initial Value	Specified In
tc-vca-channel-mo-id	—	—	—
sle-data-channel-dn	—	—	—
read-me	SLEManagementEntity (5.18)	—	—

NOTE – In accordance with the naming conventions specified for SLE data channel names in annex C, the DN for the F-tc-vca-channel has the component RDNs (mission-SC, fwd-TC-Frame, fwd-TC-VCA).

6.12.6 ACTIONS—F-tc-vca-channel MO CLASS

A F-tc-vca-channel managed object performs no actions.

6.12.7 STATE-RELATED BEHAVIOR—F-tc-vca-channel MO CLASS

A F-tc-vca-channel managed object performs no state transitions.

6.12.8 NOTIFICATIONS—F-tc-vca-channel MO CLASS

A F-tc-vca-channel managed object issues no notifications.

6.12.9 VALIDATION ASPECTS—F-tc-vca-channel MO CLASS

There are no specific validation aspects identified for this managed object class.

6.13 F-tc-virtual-channel MANAGED OBJECT CLASS

6.13.1 PURPOSE

An instance of the F-tc-virtual-channel managed object class represents the characteristics of a TC frame data channel.

6.13.2 INHERITANCE—F-tc-virtual-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.13.3 OBJECTS CONTAINED—F-tc-virtual-channel MO CLASS

An instance of this concrete managed object class shall contain zero or one instance of the F-tc-vca-channel managed object class. This class is specified in 6.12.

6.13.4 STATES—F-tc-virtual-channel MO CLASS

A F-tc-virtual-channel managed object has no states.

6.13.5 ATTRIBUTES—F-tc-virtual-channel MO CLASS

The F-tc-virtual-channel managed object class shall have the following class-specific attributes:

a) **tc-virtual-channel-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/values: Character string.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

b) **tc-fecf-flag**. Whether the optional TC transfer frame error control field is to be inserted in the transfer frames of the TC virtual channel.

   1) Data type/values:

      i) 'used'.

      ii) 'not used'.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

c) **max-frame-rate**. The maximum rate (in frames per second) that TC Frames will be carried by the data channel.

   1) Data type/values: Integer.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

d) **aggregation-allowed**. *Aggregation-allowed* specifies whether aggregation of packets is permitted for the MAPs carried by this virtual channel.

   1) Data type/values:

      i) 'permitted'.

      ii) 'not-permitted'.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

e) **max-tcfdu-length.** The *max-tcfdu-length* is the maximum length (in octets) of the TC Frame Data Unit.

   1) Data type/value: Unsigned integer.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

   NOTE – A TC Frame Data Unit can be:

   – a TC segment containing one portion of a TC User Data Unit;

   – a TC segment containing one complete TC User Data Unit;

   – a TC segment containing multiple complete TC Packets;

   – one or more complete TC Packets.

f) **cop-id.** The Command Operation Procedure (COP) currently in use on the VC. The attribute value represents the COP identifier.

   1) Data type/value: Unsigned integer. The default value shall be "1".

   2) Modifications: This attribute shall not be modifiable using the SET operation.

g) **Attribute Value Summary**. Refer to table 6-8.

Table 6-8: Attribute Value Summary—F-tc-virtual-channel Managed Object

Attribute	Inherited From	Initial Value	Specified In
tc-virtual-channel-mo-id	—	—	—
tc-fecf-flag	—	—	—
max-frame-rate	—	—	—
aggregation-allowed	—	—	—
max-tcfdu-length	—	—	—
cop-id	—	—	—
read-me	SLEManagementEntity (5.18)	—	—

6.13.6 ACTIONS—F-tc-virtual-channel MO CLASS

An F-tc-virtual-channel managed object performs no actions.

6.13.7 STATE-RELATED BEHAVIOR—F-tc-virtual-channel MO CLASS

An F-tc-virtual-channel managed object performs no state transitions.

6.13.8 NOTIFICATIONS—F-tc-virtual-channel MO CLASS

An F-tc-virtual-channel managed object issues no notifications.

6.13.9 VALIDATION ASPECTS—F-tc-virtual-channel MO CLASS

No specific validation aspects are identified for this managed object class.

6.14 F-vca-channel MANAGED OBJECT CLASS

NOTE – Although the *Cross Support Reference Model* (reference [1] specifies a set of forward AOS SLE data channels, the current suite of SLE services do not contain any forward AOS services to carry these forward AOS SLE data channels. This managed object class definition subsection is a placeholder for one that will be needed when forward AOS services are specified.

6.14.1 PURPOSE

An instance of the F-vca-channel managed object class represents the characteristics of a forward VCA data channel.

6.14.2 INHERITANCE—F-vca-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.15 R-af-channel MANAGED OBJECT CLASS

6.15.1 PURPOSE

An instance of the R-af-channel managed object class represents the characteristics of a return all frames data channel.

6.15.2 INHERITANCE—R-af-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.15.3 OBJECTS CONTAINED—R-af-channel MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero, one, or more R-master-channel managed objects. This class is specified in 6.19.

b) Zero or one R-insert-channel managed objects. This class is specified in 6.18. An R-af-channel MO may contain an R-insert-channel MO only if all frames in the RAF channel are Version 2 (AOS) frames.

6.15.4 STATES—R-af-channel MO CLASS

An R-af-channel managed object has no states.

6.15.5 ATTRIBUTES—R-af-channel MO CLASS

The R-af-channel managed object class has the following class-specific attributes:

a) **r-af-channel-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

b) **max-frame-length**. The upper bound on the length (in octets) of CCSDS frames on this all frames channel for the life of the service agreement. The frame length is measured between two synchronization words on a symbol stream, i.e., the length of the ‘encoded frame’. This corresponds to the ‘transfer frame length’ plus RS check symbols in PTM.

   1) Data type/value: Integer.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

   3) Dependencies:

i) The value of this attribute is an integer in the range [5 – 1275], inclusive, when all of the frames are V1 (Packet Telemetry) frames.

ii) If any of the frames are V2 (C/VCDU) frames, then the range is [124 – 1275] (AOS Blue Book, 5.4.10.1.1.a).

NOTE – The length of the frame does not include any Reed-Solomon check symbols that may be present.

c) **tlm-randomization-options**. Whether the use (or non-use) of pseudo-randomization on this all frames channel will be constant, or will change on a service package basis.

1) Data type/values:

i) 'always randomized' if pseudo-randomization is always present on this channel.

ii) 'never randomized' if pseudo-randomization is never present on this channel.

iii) 'package optional' if the presence of pseudo-randomization varies from service package to service package.

2) Modifications: This attribute shall not be modifiable using the SET operation.

3) Dependencies: If the value is 'package optional', a value must be assigned to the *tlm-randomization* attribute of the corresponding r-af-prod managed object contained in the servicePackage managed object.

d) **fec-options**. Whether the use (or non-use) of frame error control on this all frames channel will be constant, or will change on a service package basis.

1) Data type/values:

i) 'reed-solomon-always' if Reed-Solomon encoding (only) is to be used in all service packages.

ii) 'crc-always' if CRC encoding (only) is to be used in all service packages.

iii) 'reed-solomon&crc-always' if both Reed-Solomon and CRC encoding are to be used in all service packages.

iv) 'fec-optional' if the selection of forward error correction is deferred to the creation of the service package.

2) Modifications: This attribute shall not be modifiable using the SET operation.

3) Dependencies: If the value is 'fec-optional', a value must be assigned to the *r-fec* attribute of the corresponding r-af-prod managed object contained in the servicePackage managed object.

e) **interleave-options**. The valid depths of interleave that will be used on this all frames channel.

1) Data type/values: This is a set-valued attribute, where the possible members of the set are [1, 2, 3, 4, 5]. If the set has only member, the depth of interleave is fixed at that value for all service packages.

2) Modifications: This attribute shall not be modifiable using the SET operation.

3) Dependencies: This attribute is valid only if the value of the fec-options attribute is 'reed-solomon-always', 'reed-solomon&crc-always', or 'fec-optional'.

f) **sle-data-channel-dn**. The distinguished name of the data channel.

1) Data type/values: DN.

2) Modifications: This attribute shall not be modifiable using the SET operation.

g) **Attribute Value Summary**. Refer to table 6-9.

Table 6-9: Attribute Value Summary—R-af-channel Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-af-channel-mo-id	—	—	—
max-frame-length	—	—	—
tlm-randomization-options	—	—	—
fec-options	—	—	—
interleave-options	—	—	—
sle-data-channel-dn	—	—	—
read-me	SLEManagementEntity (5.18)	—	—

NOTE – In accordance with the naming conventions specified for SLE data channel names in annex C, the DN for the R-af-channel has the component RDNs (mission-SC, rAF).

6.15.6 ACTIONS—R-af-channel MO CLASS

A R-af-channel managed object performs no actions.

6.15.7 STATE-RELATED BEHAVIOR—R-af-channel MO CLASS

A R-af-channel managed object performs no state transitions.

6.15.8 NOTIFICATIONS—R-af-channel MO CLASS

A R-af-channel managed object issues no notifications.

6.15.9 VALIDATION ASPECTS—R-af-channel MO CLASS

If the **interleave-options** attribute has no members, then the value of the **fec-options** attribute *must* be 'crc-always'.

6.16 R-bitstream-channel MANAGED OBJECT CLASS

NOTE – Although the *Cross Support Reference Model* (reference [1]) specifies a set of forward AOS SLE data channels, the current suite of SLE services do not contain any forward AOS services to carry these forward AOS SLE data channels. This managed object class definition subsection is a placeholder for one that will be needed when forward AOS services are specified.

6.16.1 PURPOSE

An instance of the R-bitstream-channel managed object class represents the characteristics of a return space packet channel.

6.16.2 INHERITANCE—R-bitstream-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.17 R-fsh-channel MANAGED OBJECT CLASS

6.17.1 PURPOSE

An instance of the R-fsh-channel managed object class represents the characteristics of a Frame Secondary Header (FSH) data channel. The R-fsh-channel MO represents both Master Channel FSH and Virtual Channel FSH data channels, by containment by R-master-channel and R-virtual-channel MOs, respectively.

6.17.2 INHERITANCE—R-fsh-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.17.3 OBJECTS CONTAINED—R-fsh-channel MO CLASS

An instance of this concrete managed object class contains no other managed objects.

6.17.4 STATES—R-fsh-channel MO CLASS

A R-fsh-channel managed object has no states.

6.17.5 ATTRIBUTES—R-fsh-channel MO CLASS

The R-fsh-channel managed object class has the following class-specific attributes:

a) **r-fsh-channel-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

b) **max-fsh-length**. The maximum length (in octets) that the Frame Secondary Header can assume in the master channel that contains this FSH channel.

   1) Data type/value: Integer in the range [0..64].

   2) Modifications: This attribute shall not be modifiable using the SET operation.

c) **sle-data-channel-dn**. The distinguished name of the data channel.

   1) Data type/value: DN.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

d) **Attribute Value Summary**. Refer to table 6-10.

Table 6-10: Attribute Value Summary—R-fsh-channel Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-fsh-channel-mo-id	—	—	—
max-fsh-length	—	—	—
sle-data-channel-dn	—	—	—
read-me	SLEManagementEntity (5.18)	—	—

NOTE – In accordance with the naming conventions specified for SLE data channel names in annex C, the DN for the R-fsh-channel has the component RDNs (mission-SC, rAF, ret_fsh).

6.17.6 ACTIONS—R-fsh-channel MO CLASS

A R-fsh-channel managed object performs no actions.

6.17.7 STATE-RELATED BEHAVIOR—R-fsh-channel MO CLASS

A R-fsh-channel managed object performs no state transitions.

6.17.8 NOTIFICATIONS—R-fsh-channel MO CLASS

A R-fsh-channel managed object issues no notifications.

6.17.9 VALIDATION ASPECTS—R-fsh-channel MO CLASS

There are no specific validation aspects identified for this managed object class.

6.18 R-insert-channel MANAGED OBJECT CLASS

NOTE – Although the *Cross Support Reference Model* (reference [1]) specifies a set of forward AOS SLE data channels, the current suite of SLE services do not contain any forward AOS services to carry these forward AOS SLE data channels. This managed object class definition subsection is a placeholder for one that will be needed when forward AOS services are specified.

6.18.1 PURPOSE

An instance of the R-insert-channel managed object class represents the characteristics of a return insert data channel.

6.18.2 INHERITANCE—R-insert-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.19 R-master-channel MANAGED OBJECT CLASS

6.19.1 PURPOSE

An instance of the R-master-channel managed object class represents the characteristics of a return master channel.

6.19.2 INHERITANCE—R-master-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.19.3 OBJECTS CONTAINED—R-master-channel MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero, one, or more R-virtual-channel managed objects. This class is specified in 6.22.

b) Zero or one R-ocf-channel managed objects. This class is specified in 6.20.

c) Zero or one R-fsh-channel managed objects. This class is specified in 6.17.

6.19.4 STATES—R-master-channel MO CLASS

An R-master-channel managed object has no states.

6.19.5 ATTRIBUTES—R-master-channel MO CLASS

The R-master-channel managed object class has the following class-specific attributes:

a) **r-master-channel-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

b) **max-frame-rate**. The maximum rate (in frames per second) that MC Frames will be carried by the data channel.

   1) Data type/value: Integer.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

c) **sle-data-channel-dn.** The distinguished name of the data channel.

   1) Data type/value: DN.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

d) **Attribute Value Summary**. Refer to table 6-11.

Table 6-11: Attribute Value Summary—R-master-channel Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-master-channel-mo-id	—	—	—
max-frame-rate	—	—	—
sle-data-channel-dn	—	—	—
read-me	SLEManagementEntity (5.18)	—	—

NOTE – In accordance with the naming conventions specified for SLE data channel names in annex C, the DN for the R-master-channel has the component RDNs (mission-SC, rAF, ret_CF).

6.19.6 ACTIONS—R-master-channel MO CLASS

A R-master-channel managed object performs no actions.

6.19.7 STATE-RELATED BEHAVIOR—R-master-channel MO CLASS

A R-master-channel managed object performs no state transitions.

6.19.8 NOTIFICATIONS—R-master-channel MO CLASS

A R-master-channel managed object issues no notifications.

6.19.9 VALIDATION ASPECTS—R-master-channel MO CLASS

There are no specific validation aspects identified for this managed object class.

6.20 R-ocf-channel MANAGED OBJECT CLASS

6.20.1 PURPOSE

An instance of the R-ocf-channel managed object class represents the characteristics of an Operational Control Field (OCF) data channel. The R-ocf-channel MO represents both Master Channel OCF and Virtual Channel OCF data channels, by containment by R-master-channel and R-virtual-channel MOs, respectively.

6.20.2 INHERITANCE—R-ocf-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.20.3 OBJECTS CONTAINED—R-ocf-channel MO CLASS

An instance of this concrete managed object class contains no other managed objects.

6.20.4 STATES—R-ocf-channel MO CLASS

A R-ocf-channel managed object has no states.

6.20.5 ATTRIBUTES—R-ocf-channel MO CLASS

The R-ocf-channel managed object class has the following class-specific attributes:

a) **r-ocf-channel-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

b) **ocf-type-options**. The OCF types that may appear on this OCF data channel during the lifetime of the Service Agreement.

   1) Data type/value: This attribute is a set-valued attribute, where the set must contain at least one of the following values:

      i) 'ocf-type-1', indicating CLCW-bearing OCFs.

      ii) 'ocf-type-2-0', indicating project-specific-information-bearing OCFs.

iii) 'ocf-type-2-1', indicating OCFs containing to-be-defined information.

NOTE – OCF Type 2-1 is reserved for future definition by CCSDS.

2) Modifications: This attribute shall not be modifiable using the SET operation.

c) **sle-data-channel-dn.** The distinguished name of the data channel.

1) Data type/value: DN.

2) Modifications: This attribute shall not be modifiable using the SET operation.

d) **clcw-timeout**. The *clcw-timeout* is the time (in seconds) during which no CLCWs are received that must elapse in order for the status of RF lock on the forward link to be considered unknown.

1) Data type/value: positive integer in the range [0..3600].

2) Modifications: This attribute shall not be modifiable using the SET operation.

e) **Attribute Value Summary**. Refer to table 6-12.

Table 6-12: Attribute Value Summary—R-ocf-channel Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-ocf-channel-mo-id	—	—	—
ocf-type-options	—	—	—
sle-data-channel-dn	—	—	—
clcw-timeout	—	—	—
read-me	SLEManagementEntity (5.18)	—	—

NOTE – In accordance with the naming conventions specified for SLE data channel names in annex C, the DN for the R-ocf-channel has the component RDNs (mission-SC, rAF, ret_ocf).

6.20.6 ACTIONS—R-ocf-channel MO CLASS

A R-ocf-channel managed object performs no actions.

6.20.7 STATE-RELATED BEHAVIOR—R-ocf-channel MO CLASS

A R-ocf-channel managed object performs no state transitions.

6.20.8 NOTIFICATIONS—R-ocf-channel MO CLASS

clcw timeout exceeded. The *clcw-timeout-exceeded* notification indicates that no CLCWs have been received on this return OCF channel for at least as long as the period specified by the *clcw-timeout* attribute.

6.20.9 VALIDATION ASPECTS—R-ocf-channel MO CLASS

There are no specific validation aspects identified for this managed object class.

6.21 R-sp-channel MANAGED OBJECT CLASS

6.21.1 PURPOSE

An instance of the R-sp-channel managed object class represents the characteristics of a return space packet channel.

6.21.2 INHERITANCE—R-sp-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.21.3 OBJECTS CONTAINED—R-sp-channel MO CLASS

An instance of this concrete managed object class contains no other managed objects.

6.21.4 STATES—R-sp-channel MO CLASS

An R-sp-channel managed object has no states.

6.21.5 ATTRIBUTES—R-sp-channel MO CLASS

The R-sp-channel managed object class has the following class-specific attributes:

a) **r-space-packet-channel-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

b) **sle-data-channel-dn.** The distinguished name of the data channel.

   1) Data type/value: DN.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

c) **Attribute Value Summary**. Refer to table 6-13.

Table 6-13: Attribute Value Summary—R-sp-channel Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-space-packet-channel-mo-id	—	—	—
sle-data-channel-dn	—	—	—
read-me	SLEManagementEntity (5.18)	—	—

NOTE – In accordance with the naming conventions specified for SLE data channel names in annex C, the DN for the R-sp-channel has the component RDNs (mission-SC, rAF, ret_CF, ret_space-pkt).

6.21.6 ACTIONS—R-sp-channel MO CLASS

An R-sp-channel managed object performs no actions.

6.21.7 STATE-RELATED BEHAVIOR—R-sp-channel MO CLASS

An R-sp-channel managed object performs no state transitions.

6.21.8 NOTIFICATIONS—R-sp-channel MO CLASS

An R-sp-channel managed object issues no notifications.

6.21.9 VALIDATION ASPECTS—R-sp-channel MO CLASS

There are no specific validation aspects identified for this managed object class.

6.22 R-virtual-channel MANAGED OBJECT CLASS

6.22.1 PURPOSE

An instance of the R-virtual-channel managed object class represents the characteristics of a return virtual channel.

6.22.2 INHERITANCE—R-virtual-channel MO CLASS

This managed object class is derived from and inherits the properties of the SLEManagementEntity managed object class.

6.22.3 OBJECTS CONTAINED—R-virtual-channel MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero, one, or more R-sp-channel managed objects. This class is specified in 6.21. An R-virtual-channel managed object may contain R-sp-channel managed objects only if the R-virtual-channel managed object *does not* contain an R-BitstreamChannel managed object.

b) Zero or one R-ocf-channel managed objects. This class is specified in 6.20. An R-virtual-channel managed object may contain an R-ocf-channel managed object only if the R-master-channel managed object that contains the R-virtual-channel managed object *does not* contain an R-ocf-channel managed object.

c) Zero or one R-fsh-channel managed objects. This class is specified in 6.17. An R-virtual-channel managed object may contain an R-fsh-channel managed object only if

   1) the R-master-channel managed object that contains the R-virtual-channel managed object *does not* contain an R-fsh-channel managed object; *and*

   2) the R-virtual-channel is a Packet Telemetry (Version 1) virtual channel.

d) Zero or one R-BitstreamChannel managed objects. This class is specified in 6.16. An R-virtual-channel managed object may contain an R-BitstreamChannel managed object only if

   1) the R-virtual-channel managed object does not contain any R-sp-channel managed objects; and

   2) the R-virtual-channel is an AOS (Version 2) virtual channel.

6.22.4 STATES—R-virtual-channel MO CLASS

An R-virtual-channel managed object has no states.

6.22.5 ATTRIBUTES—R-virtual-channel MO CLASS

The R-virtual-channel managed object class has the following class-specific attributes:

a) **r-virtual-channel-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

b) **max-frame-rate**. The maximum rate (in frames per second) that VC frames will be carried by the data channel.

   1) Data type/value: Integer.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

c) **sle-data-channel-dn**. The distinguished name of the data channel.

   1) Data type/value: DN.

   2) Modifications: This attribute shall not be modifiable using the SET operation.

d) **Attribute Value Summary**. Refer to table 6-14.

Table 6-14: Attribute Value Summary—R-virtual-channel Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-virtual-channel-mo-id	—	—	—
max-frame-rate	—	—	—
sle-data-channel-dn	—	—	—
read-me	SLEManagementEntity (5.18)	—	—

NOTE – In accordance with the naming conventions specified for SLE data channel names in annex C, the DN for the R-virtual-channel has the component RDNs (mission-SC, rAF, ret_CF).

6.22.6 ACTIONS—R-virtual-channel MO CLASS

An R-virtual-channel managed object performs no actions.

6.22.7 STATE-RELATED BEHAVIOR—R-virtual-channel MO CLASS

An R-virtual-channel managed object performs no state transitions.

6.22.8 NOTIFICATIONS—R-virtual-channel MO CLASS

An R-virtual-channel managed object issues no notifications.

6.22.9 VALIDATION ASPECTS—R-virtual-channel MO CLASS

There are no specific validation aspects identified for this managed object class.

7 RETURN SPACE LINK PROCESSING MANAGED OBJECT DEFINITIONS

7.1 R-af-prod MANAGED OBJECT CLASS

7.1.1 PURPOSE

This managed object represents the processing associated with consuming a physical channel (baseband stream of bits) and processing it to form an rAF channel (frame sync, error detection and annotation, header annotation, notification generation, etc.). This managed object is executed as part of the Return Space Link Processing functional group.

7.1.2 INHERITANCE—R-af-prod MO CLASS

This managed object class is derived from and inherits the properties of the Production managed object class.

7.1.3 OBJECTS CONTAINED—R-af-prod MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero, one, or more r-af-st managed objects. This class is specified in 7.2.

b) Zero, one, or more r-af-ts-p managed objects. This class is specified in 7.3.

c) Zero or one r-insert-prod managed objects. This class is specified in 7.4.

d) Zero or one r-cf-prod managed objects. This class is specified in 8.2.

7.1.4 STATES—R-af-prod MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the Production managed object class.

7.1.5 ATTRIBUTES—R-af-prod MO CLASS

This managed object class has the following specific attributes:

a) **r-af-prod-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **r-fec.** The frame error control used on the r-af-channel.

   1) Data type/values:

i) 'Reed-Solomon'. Only Reed-Solomon coding is used.

ii) 'CRC'. Only CRC error detection (using the Frame Error Control Field (V1 frames) or VCDU Error Control Field (V2 frames)) is used.

iii) 'Reed-Solomon+CRC'. Both types of error control are present.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **r-frame-length.** The length (in octets) of the frames on the RAF channel to be processed. The length of the Attached Sync Marker and Reed-Solomon check symbols (if Reed-Solomon coding is used) are <u>not</u> included in this value.

1) Data type/value: Integer:

i) The upper bound on the value is specified by the value of *max-frame-length* in the corresponding r-af-channel managed object.

ii) The lower bound on the value is 5 when all of the VCs are V1 (Packet Telemetry) VCs, and 124 if any of the VCs are V2 (C/VCDU) VCs (AOS Blue Book, reference [5], subsection 5.4.10.1.1.a).

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

d) **tlm-randomization**. This attribute specifies whether the Pseudo-Randomizer (see section 6 of *Telemetry Channel Coding,* reference [4]) will be applied to the r-af-channel to be received.

1) Data type/values:

i) 'always randomized', indicating that randomization will be applied.

ii) 'never randomized', indicating that randomization will not be applied.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

3) Dependencies: This attribute must be specified in the r-af-prod managed object if the value of the *tlm-randomization-options* attribute in the corresponding r-af-channel managed object is 'package optional'. If the value of the corresponding *tlm-randomization-options* attribute is either 'always randomized' or 'never randomized', the value of the *tlm-randomization* attribute need not be specified. However, if it is specified, it must match the value of the corresponding *tlm-randomization-options* attribute or a validation error shall result.

e) **interleave-depth**. The depth of interleave that will be used on this all frames channel during the space link session.

1) Data type/values: This attribute has an integer syntax, with value equal to one of the values in the set contained in the *interleave-options* attribute of the corresponding r-af-channel managed object.

2) Modifications: SET shall be enabled when the managed object is in the state *validated* or *committed*.

3) Dependencies: This attribute is valid only if the value of *r-fec* is 'Reed-Solomon' or 'Reed-Solomon+CRC'.

NOTE – Modifications made while in the *committed* state may result in temporary loss of synchronization and data.

f) **allowed-sle-channel-dn.** The *allowed-sle-channel-dn* identifies the SLE data channel that is handled by the SLE resource represented by this managed object.

1) Data type/value: DN. This name must be formed according to the rules defined in annex C.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

g) **Attribute Value Summary**. Refer to table 7-1.

Table 7-1: Attribute Value Summary—R-af-prod Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-af-prod-mo-id	—	by CM	CREATE.inv
r-fec	—	by UM	CREATE.inv
r-frame-length	—	by UM	CREATE.inv
tlm-randomization	—	by UM	CREATE.inv
interleave-depth	—	by UM	CREATE.inv
allowed-sle-channel-dn	—	by UM	CREATE.inv
production-state	Production (5.10)	= *waiting*	—
data-units-consumed-for-processing	Production (5.10)	= 0	as needed
processed-successfully	Production (5.10)	= 0	as needed
data-units-produced	Production (5.10)	= 0	as needed
number-invocations	Production (5.10)	= 0	as needed
unavailable-time	Production (5.10)	= 0	as needed
operational-time	Production (5.10)	= 0	as needed
production-start-time	Production (5.10)	by CM	when production starts
production-stop-time	Production (5.10)	by CM	when production stops
done-time	Production (5.10)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

7.1.6 ACTIONS—R-af-prod MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the Production managed object class.

7.1.7 STATE-RELATED BEHAVIOR—R-af-prod MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the Production managed object class.

7.1.8 NOTIFICATIONS—R-af-prod MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the Production managed object class.

7.1.9 VALIDATION ASPECTS—R-af-prod MO CLASS

a) The name of the channel produced shall be appropriate for an instance of return all frames service.

b) The *allowed-sle-channel-dn* must exist in the channel tree and must match the channel type that the present managed object can process. If the respective channel cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

c) When an r-af-prod managed object contains an r-af-st managed object, the values of the *allowed-sle-channel-dn* attributes of the two managed objects must be equal.

d) The attribute *interleave-depth* must be included in the CREATE invocation for an r-af-prod managed object if the set value of *interleave-options* has more than one member in the corresponding r-af-channel managed object. The attribute *interleave-depth* may be included in the CREATE invocation for the r-af-prod managed object if the value for *interleave-options* has only one member, but the value of *interleave-depth* must be set to the value of the single member of *interleave-options.*

e) Once the r-af-prod managed object has been created, the value of *interleave-depth* may be changed only if the value of the corresponding *interleave options* set has more than one value. It may be reset only to one of the values in the set.

f) The attribute r-fec must be included in the CREATE invocation for the r-af-prod managed object if the value for *fec-options* is 'fec-optional' in the corresponding r-af-channel managed object. The attribute *r-fec* may be included in the CREATE invocation for the r-af-prod managed object if the value for *fec-options* is 'reed-solomon-always', 'crc-always', or 'reed-solomon&crc-always', but the value of *r-fec* must be 'Reed-Solomon', 'CRC', or 'Reed-Solomon+CRC', respectively.

g) Once the r-af-prod managed object has been created, the value of *r-fec* may be modified only if the value of *fec-options* is 'fec-optional' in the corresponding r-af-channel managed object. If the value of *fec-options* is 'reed-solomon-always', 'crc-always', or 'reed-solomon&crc-always', the value of *r-fec* is fixed to 'Reed-Solomon', 'CRC', or 'Reed-Solomon+CRC', respectively, and cannot be modified.

7.2 R-af-st MANAGED OBJECT CLASS

7.2.1 PURPOSE

This managed object represents the resources needed to store an RAF channel. Refer to 5.3 for general storage considerations.

7.2.2 INHERITANCE—R-af-st MO CLASS

This managed object class is derived from and inherits the properties of the ChannelStored managed object class.

7.2.3 OBJECTS CONTAINED—R-af-st MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero, one, or more r-af-ts-p managed objects. This class is specified in 7.3.

b) Zero or one r-cf-prod managed objects. This class is specified in 8.2.

7.2.4 STATES—R-af-st MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the ChannelStored managed object class.

7.2.5 ATTRIBUTES—R-af-st MO CLASS

This managed object class has the following specific attributes:

a) **r-af-st-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string with a maximum length of 256 ASCII characters.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **allowed-sle-channel-dn**. The SLE data channel that is handled by the SLE resource represented by this managed object.

   1) Data type/value: DN. This name must be formed according to the rules defined in annex C.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **Attribute Value Summary**. Refer to table 7-2.

Table 7-2: Attribute Value Summary—R-af-st Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-af-st-mo-id	—	by CM	CREATE.inv
allowed-sle-channel-dn	—	by UM	CREATE.inv
data-store-id	ChannelStored (5.3)	by UM	CREATE.inv
partition-id	ChannelStored (5.3)	by UM	CREATE.inv
access-mode	ChannelStored (5.3)	by UM	CREATE.inv
ingest-window-start-time	ChannelStored (5.3)	by UM	CREATE.inv
ingest-window-stop-time	ChannelStored (5.3)	by UM	CREATE.inv
ingest-time	ChannelStored (5.3)	= 0	CREATE.rtn
retrieval-window-start-time	ChannelStored (5.3)	by UM	CREATE.inv
retrieval-window-stop-time	ChannelStored (5.3)	by UM	CREATE.inv
unavailable-time	ChannelStored (5.3)	= 0	CREATE.rtn
provision-time	ChannelStored (5.3)	= 0	CREATE.rtn
number-SLE-PDUs-stored	ChannelStored (5.3)	= 0	CREATE.rtn
channel-stored-state	ChannelStored (5.3)	= *waiting*	CREATE.inv
done-time	ChannelStored (5.3)	by CM	when MO enters *done*
partition-data-units	ChannelStored (5.3)	= 0	as needed
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

7.2.6 ACTIONS—R-af-st MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the ChannelStored managed object class.

7.2.7 STATE-RELATED BEHAVIOR—R-af-st MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the ChannelStored managed object class.

7.2.8 NOTIFICATIONS—R-af-st MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the ChannelStored managed object class.

7.2.9 VALIDATION ASPECTS—R-af-st MO CLASS

a) At least one managed object shall be contained. If not, a ‘consistency defect’ shall be added to the containing servicePackage managed object's set of *defect-records*.

b) The name of the channel stored shall be appropriate for an instance of return all frames service.

c) The *allowed-sle-channel-dn* must exist in the channel tree and must match the channel type that the present managed object can process. If the respective channel cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

7.3 R-af-ts-p MANAGED OBJECT CLASS

7.3.1 PURPOSE

This managed object represents the resources needed to provide an RAF transfer service. Refer to 5.20 for a general discussion of transfer service provision.

7.3.2 INHERITANCE—R-af-ts-p MO CLASS

This managed object class is derived from and inherits the properties of the R-transferService managed object class.

7.3.3 OBJECTS CONTAINED—R-af-ts-p MO CLASS

An instance of this concrete managed object class contains no other managed objects.

7.3.4 STATES—R-af-ts-p MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the R-transferService managed object class.

7.3.5 ATTRIBUTES—R-af-ts-p MO CLASS

This managed object class has the following specific attributes:

a) **r-af-ts-p-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string with a maximum length of 256 ASCII characters.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **allowed-frame-quality.** The set of allowed frame quality.

   1) Data type/values:

      i) 'good-frames-only'.

      ii) 'erred-frames-only'.

      iii) 'all-frames'.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **errored-RAF-PDUs.** The number of incomplete or errored RAF-PDUs encountered.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

d) **RAF-data-units-supplied**. The number of RAF channel data units supplied by the service instance.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

e) **RAF-data-units-available**. The number of RAF channel data units available, but not supplied by the service instance due to timely delivery requirements.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

f) **RAF-PDUs-supplied**. The number of RAF-PDUs supplied by the service instance.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

g) **frames-sent**. The number of frames sent.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase.

      i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

NOTE – There is no attribute specifying the DN of the RAF channel carried by this service instance. The DN of the RAF channel is shared with this managed object by the r-af-prod managed object that contains it. The value of the DN is contained in the *allowed-sle-channel-dn* attribute of the r-af-prod managed object.

h) **Attribute Value Summary**. Refer to table 7-3.

Table 7-3: Attribute Value Summary—R-af-ts-p Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-af-ts-p-mo-id	—	by CM	CREATE.inv
allowed-frame-quality	—	by UM	CREATE.inv
errored-raf-pdus	—	= 0	as needed
raf-data-units-supplied	—	= 0	as needed
raf-data-units-available	—	= 0	as needed
raf-pdus-supplied	—	= 0	as needed
frames-sent	—	= 0	as needed
r-delivery-mode	R-transferService (5.15)	by UM	CREATE.inv
delivery-quality	R-transferService (5.15)	by UM	CREATE.inv
transfer-buffer-size	R-transferService (5.15)	by UM	CREATE.inv
latency-limit	R-transferService (5.15)	by UM	CREATE.inv
max-number-concatenations	R-transferService (5.15)	by UM	CREATE.inv
current-ts-version	TransferService (5.20)	by UM	CREATE.inv
lower-bound-reporting-period	TransferService (5.20)	by UM	CREATE.inv
time-out-period	TransferService (5.20)	by UM	CREATE.inv
transfer-service-state	TransferService (5.20)	= *waiting*	no
association-initiation-responsibility	TransferService (5.20)	by UM	CREATE.inv
responder-port-identifier	TransferService (5.20)	by UM	CREATE.inv
local-sle-entity-identifier	TransferService (5.20)	by CM	CREATE.inv
peer-sle-entity-access-list	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-start-time	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-stop-time	TransferService (5.20)	by UM	CREATE.inv
first-data-unit-time	TransferService (5.20)	by CM	when first data unit is transferred
last-data-unit-time	TransferService (5.20)	by CM	when last data unit is transferred
service-instance-start	TransferService (5.20)	= 'null'	when first service instance starts
service-instance-stop	TransferService (5.20)	= 'null'	when last service instance stops
provision-time	TransferService (5.20)	= 0	as needed
failed-binds	TransferService (5.20)	= 0	as needed
unavailable-time	TransferService (5.20)	= 0	as needed
done-time	TransferService (5.20)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

7.3.6 ACTIONS—R-af-ts-p MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the R-transferService managed object class.

7.3.7 STATE-RELATED BEHAVIOR—R-af-ts-p MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the R-transferService managed object class.

7.3.8 NOTIFICATIONS—R-af-ts-p MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the R-transferService managed object class.

7.3.9 VALIDATION ASPECTS—R-af-ts-p MO CLASS

a) The name of the channel provided shall be appropriate for an instance of return all frames service.

b) The *allowed-sle-channel-dn* must exist in the channel tree and must match the channel type that the present managed object can process. If the respective channel cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records.*

7.4 R-insert-prod MANAGED OBJECT CLASS

NOTE – This managed object supports the provision of r-insert services, which at the time of approval of this draft Recommendation have been identified in *the Cross Support Reference Model* (reference [1]), but has not yet been specified. Therefore, this managed object definition is currently for informative purposes only. When the service specification is developed and approved, this managed object definition will be updated as necessary to correspond to the as-specified SLE service, after which this managed object definition will become normative.

7.4.1 PURPOSE

The R-insert-prod managed object class represents the return insert service. This service is the processing associated with extracting an insert zone from each frame in an RAF channel and processing it to form an insert channel. One r-insert-prod managed object exists for each RAF channel that contains an insert zone to be extracted and provided via one or more instances of online or offline return insert service.

An r-insert-prod managed object consumes one RAF channel and produces one R-insert channel.

7.4.2 INHERITANCE—R-insert-prod MO CLASS

This managed object class is derived from and inherits the properties of the Production managed object class.

7.5 R-insert-st MANAGED OBJECT CLASS

NOTE – This managed object supports the provision of r-insert services, which at the time of approval of this draft Recommendation have been identified in the *Cross Support Reference Model* (reference [1]), but has not yet been specified. Therefore, this managed object definition is currently for informative purposes only. When the service specification is developed and approved, this managed object definition will be updated as necessary to correspond to the as-specified SLE service, after which this managed object definition will become normative.

7.5.1 PURPOSE

This managed object class represents the resources needed to store a return insert channel. Refer to 5.3 for general storage considerations.

7.5.2 INHERITANCE—R-insert-st MO CLASS

This managed object class is derived from and inherits the properties of the ChannelStored managed object class.

7.6 R-insert-ts-p MANAGED OBJECT CLASS

NOTE – This managed object supports the provision of r-insert services, which at the time of approval of this draft Recommendation have been identified in the *Cross Support Reference Model* (reference [1]) but has not yet been specified. Therefore, this managed object definition is currently for informative purposes only. When the service specification is developed and approved, this managed object definition will be updated as necessary to correspond to the as-specified SLE service, after which this managed object definition will become normative.

7.6.1 PURPOSE

This managed object represents the resources needed to provide a return insert transfer service. Refer to 5.20 for a general discussion of transfer service provision.

7.6.2 INHERITANCE—R-insert-ts-p MO CLASS

This managed object class is derived from and inherits the properties of the R-transferService managed object class.

8 RETURN FRAME PROCESSING MANAGED OBJECT DEFINITIONS

8.1 R-af-ts-u MANAGED OBJECT CLASS

8.1.1 PURPOSE

This managed object represents the resources needed to use an RAF transfer service. Refer to 5.20 for a general discussion of transfer service use.

8.1.2 INHERITANCE—R-af-ts-u MO CLASS

This managed object class is derived from and inherits the properties of the R-transferService managed object class.

8.1.3 OBJECTS CONTAINED—R-af-ts-u MO CLASS

An instance of this concrete managed object class must contain at least one instance of either (or both) of the following classes:

a) zero or one r-cf-prod managed object. This class is specified in 8.2.

b) zero, one, or more r-af-st managed objects. This class is specified in 7.2.

8.1.4 STATES—R-af-ts-u MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the R-transferService managed object class.

8.1.5 ATTRIBUTES—R-af-ts-u MO CLASS

This managed object class has the following specific attributes:

a) **r-af-ts-u-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string with a maximum length of 256 ASCII characters.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **allowed-sle-channel-dn.** The SLE data channel that is handled by the SLE resource represented by this managed object.

   1) Data type/value: DN. This name must be formed according to the rules defined in annex C.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **service-instance-identifier.** The DN of the service instance to be used.

   1) Data type/value: The value is equal to the DN of the transfer 'service provided' managed object within the service-providing SLE Complex associated with the desired service instance.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

d) **frames-consumed**. The number of frames consumed.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

e) **RAF-data-units-consumed**. The number of RAF channel data units consumed by the service instance.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

f) **RAF-PDUs-consumed**. The number of RAF-PDUs consumed by the service instance.

   1) Data type/values: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

g) **ert-interval-begin**. The value to be placed in the start-time parameter of the START invocation.

   1) Data type/value: CCSDS time code.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

3) Dependencies: This attribute is applicable only if the value of *r-delivery-mode* is 'return offline'.

h) **ert-interval-end**. The value to be placed in the stop-time parameter of the START invocation.

1) Data type/value: CCSDS time code.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting.*

3) Dependencies: This attribute is applicable only if the value of *r-delivery-mode* is 'return offline'.

i) **Attribute Value Summary**. Refer to table 8-1.

Table 8-1: Attribute Value Summary—R-af-ts-u Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-af-ts-u-mo-id	—	by CM	CREATE.inv
allowed-sle-channel-dn	—	by UM	CREATE.inv
service-instance-identifier	—	by CM	CREATE.inv
frames-consumed	—	= 0	as needed
raf-data-units-consumed	—	= 0	as needed
raf-pdus-consumed	—	= 0	as needed
ert-interval-begin	—	by UM	CREATE.inv
ert-interval-end	—	by UM	CREATE.inv
r-delivery-mode	R-transferService (5.15)	by UM	CREATE.inv
delivery-quality	R-transferService (5.15)	by UM	CREATE.inv
transfer-buffer-size	R-transferService (5.15)	by UM	CREATE.inv
latency-limit	R-transferService (5.15)	by UM	CREATE.inv
max-number-concatenations	R-transferService (5.15)	by UM	CREATE.inv
current-ts-version	TransferService (5.20)	by UM	CREATE.inv
lower-bound-reporting-period	TransferService (5.20)	by UM	CREATE.inv
time-out-period	TransferService (5.20)	by UM	CREATE.inv
transfer-service-state	TransferService (5.20)	= *waiting*	no
association-initiation-responsibility	TransferService (5.20)	by UM	CREATE.inv
responder-port-identifier	TransferService (5.20)	by UM	CREATE.inv
local-sle-entity-identifier	TransferService (5.20)	by CM	CREATE.inv
peer-sle-entity-access-list	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-start-time	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-stop-time	TransferService (5.20)	by UM	CREATE.inv
first-data-unit-time	TransferService (5.20)	by CM	when first data unit is transferred
last-data-unit-time	TransferService (5.20)	by CM	when last data unit is transferred
service-instance-start	TransferService (5.20)	= 'null'	when first service instance starts
service-instance-stop	TransferService (5.20)	= 'null'	when last service instance stops
provision-time	TransferService (5.20)	= 0	as needed
failed-binds	TransferService (5.20)	= 0	as needed
unavailable-time	TransferService (5.20)	= 0	as needed
done-time	TransferService (5.20)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

8.1.6 ACTIONS—R-af-ts-u MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the R-transferService managed object class.

8.1.7 STATE-RELATED BEHAVIOR—R-af-ts-u MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the R-transferService managed object class.

8.1.8 NOTIFICATIONS—R-af-ts-u MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the R-transferService managed object class.

8.1.9 VALIDATION ASPECTS—R-af-ts-u MO CLASS

a) At least one managed object shall be contained. If not, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

b) The name of the channel used shall be appropriate for an instance of return all frames service.

c) The *allowed-sle-channel-dn* must exist in the channel tree and must match the channel type that the present managed object can process. If the channel cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

8.2 R-cf-prod MANAGED OBJECT CLASS

8.2.1 PURPOSE

This managed object embodies the management interface to processes that perform the demultiplexing of frame channels from a RAF data channel. This production is executed as part of the Return Frame Processing functional group.

8.2.2 INHERITANCE—R-cf-prod MO CLASS

This managed object class is derived from and inherits the properties of the Production managed object class.

8.2.3 OBJECTS CONTAINED—R-cf-prod MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero, one, or more r-cf-ts-p managed objects. This class is specified in 8.4.

b) Zero, one, or more r-cf-st managed objects. This class is specified in 8.3.

c) Zero or one r-fsh-prod managed objects. Presence of an instance is mandatory if the Complex stores FSH channels or provides FSH service as part of this Service Package. This class is specified in 8.5.

d) Zero or one r-ocf-prod managed objects. This class is specified in 8.8. Presence of an instance is mandatory if the Complex

   1) stores OCF channels;

   2) provides OCF service as part of this Service Package; or

   3) implements an instance of r-cf-prod and either (or both) of the f-tc-session-prod or f-tc-vc-prod managed objects.

e) Zero or one r-frameDataField-prod managed object. Presence of an instance is mandatory if the Complex stores space packet channels or bitstream channels, or provides space packet or bitstream service. This class is specified in 9.4.

8.2.4 STATES—R-cf-prod MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the Production managed object class.

8.2.5 ATTRIBUTES—R-cf-prod MO CLASS

This managed object class has the following specific attribute:

a) **r-cf-prod-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **Attribute Value Summary**. Refer to table 8-2.

Table 8-2: Attribute Value Summary—R-cf-prod Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-cf-prod-mo-id	—	by CM	CREATE.inv
production-state	Production (5.10)	= *waiting*	—
data-units-consumed-for-processing	Production (5.10)	= 0	as needed
processed-successfully	Production (5.10)	= 0	as needed
data-units-produced	Production (5.10)	= 0	as needed
number-invocations	Production (5.10)	= 0	as needed
unavailable-time	Production (5.10)	= 0	as needed
operational-time	Production (5.10)	= 0	as needed
production-start-time	Production (5.10)	by CM	when production starts
production-stop-time	Production (5.10)	by CM	when production stops
done-time	Production (5.10)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

8.2.6 ACTIONS—R-cf-prod MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the Production managed object class.

8.2.7 STATE-RELATED BEHAVIOR—R-cf-prod MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the Production managed object class.

8.2.8 NOTIFICATIONS—R-cf-prod MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the Production managed object class.

8.2.9 VALIDATION ASPECTS—R-cf-prod MO CLASS

A 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records* if the managed object does not contain an instance of r-ocf-prod and the Complex

a) stores OCF channels; or

b) provides OCF service as part of this Service Package; or

c) implements an instance of r-cf-prod <u>and</u> either (or both) of the f-tc-session-prod or f-tc-vc-prod managed objects.

8.3 R-cf-st MANAGED OBJECT CLASS

8.3.1 PURPOSE

This managed object represents the resources needed to store a Return Frame channel. Refer to 5.3 for general storage considerations.

8.3.2 INHERITANCE—R-cf-st MO CLASS

This managed object class is derived from and inherits the properties of the ChannelStored managed object class.

8.3.3 OBJECTS CONTAINED—R-cf-st MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero, one, or more r-cf-ts-p managed objects. Presence of an instance is mandatory if the Complex provides return channel frame service(s) as part of the service package. This class is specified in 8.4.

b) Zero or one r-frameDataField-prod managed object. This class is specified in 9.4.

8.3.4 STATES—R-cf-st MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the ChannelStored managed object class.

8.3.5 ATTRIBUTES—R-cf-st MO CLASS

This managed object class has the following specific attributes:

a) **r-cf-st-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string with a maximum length of 256 ASCII characters.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **stored-channel-frame-dns.** The frame channels that are stored by the storage resource represented by this managed object.

   1) Data type/values: Set of DNs. These names must be formed according to the rules defined in annex C.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **Attribute Value Summary**. Refer to table 8-3.

Table 8-3: Attribute Value Summary—R-cf-st Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-cf-st-mo-id	—	by CM	CREATE.inv
stored-channel-frame-dns	—	by UM	CREATE.inv
data-store-id	ChannelStored (5.3)	by UM	CREATE.inv
partition-id	ChannelStored (5.3)	by UM	CREATE.inv
access-mode	ChannelStored (5.3)	by UM	CREATE.inv
ingest-window-start-time	ChannelStored (5.3)	by UM	CREATE.inv
ingest-window-stop-time	ChannelStored (5.3)	by UM	CREATE.inv
ingest-time	ChannelStored (5.3)	=0	CREATE.rtn
retrieval-window-start-time	ChannelStored (5.3)	by UM	CREATE.inv
retrieval-window-stop-time	ChannelStored (5.3)	by UM	CREATE.inv
unavailable-time	ChannelStored (5.3)	= 0	CREATE.rtn
provision-time	ChannelStored (5.3)	= 0	CREATE.rtn
number-SLE-PDUs-stored	ChannelStored (5.3)	= 0	CREATE.rtn
channel-stored-state	ChannelStored (5.3)	= *waiting*	CREATE.inv
done-time	ChannelStored (5.3)	by CM	when MO enters *done*
partition-data-units	ChannelStored (5.3)	= 0	as needed
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

8.3.6 ACTIONS—R-cf-st MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the ChannelStored managed object class.

8.3.7 STATE-RELATED BEHAVIOR—R-cf-st MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the ChannelStored managed object class.

8.3.8 NOTIFICATIONS—R-cf-st MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the ChannelStored managed object class.

8.3.9 VALIDATION ASPECTS—R-cf-st MO CLASS

a) At least one managed object shall be contained. If not, a ‘consistency defect’ shall be added to the containing servicePackage managed object's set of *defect-records*.

b) The name of the channel stored shall be appropriate for an instance of return channel frames service.

c) The *stored-channel-frame-dns* must exist in the channel tree and must match the channel type that the present managed object can process. If the channels cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

8.4 R-cf-ts-p MANAGED OBJECT CLASS

8.4.1 PURPOSE

This managed object represents the resources needed to provide a Return Channel Frame (RCF) service. Refer to 5.20 for a general discussion of transfer service provision.

8.4.2 INHERITANCE—R-cf-ts-p MO CLASS

This managed object class is derived from and inherits the properties of the R-transferService managed object class.

8.4.3 OBJECTS CONTAINED—R-cf-ts-p MO CLASS

An instance of this concrete managed object class contains no other managed objects.

8.4.4 STATES—R-cf-ts-p MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the R-transferService managed object class.

8.4.5 ATTRIBUTES—R-cf-ts-p MO CLASS

This managed object class has the following specific attributes:

a) **r-cf-ts-p-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string with a maximum length of 256 ASCII characters.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **allowed-channel-frame-dns.** The set of DNs of the SLE data channels that may be handled by the SLE resource represented by this managed object during the service instance life time.

   1) Data type/values: Set of DNs. These names must be formed according to the rules defined in annex C.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **RCF-data-units-sent**. The number of RCF data units sent.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

d) **RCF-data-units-supplied**. The number of RCF data units supplied by the service instance.

1) Data type/value: Unsigned integer. Initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase.

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

e) **RCF-data-units-available**. The number of RCF data units available, but not supplied by the service instance due to timely delivery requirements.

1) Data type/value: Unsigned integer. Initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase.

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

f) **RCF-PDUs-supplied**. The number of RCF-PDUs supplied by the service instance.

1) Data type/value: Unsigned integer. Initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

g) **Attribute Value Summary**. Refer to table 8-4.

Table 8-4: Attribute Value Summary—R-cf-ts-p Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-cf-ts-p-mo-id	—	by CM	CREATE.inv
allowed-channel-frame-dns	—	by UM	CREATE.inv
rcf-data-units-sent	—	= 0	as needed
rcf-data-units-supplied	—	= 0	as needed
rcf-data-units-available	—	= 0	as needed
rcf-pdus-supplied	—	= 0	as needed
r-delivery-mode	R-transferService (5.15)	by UM	CREATE.inv
delivery-quality	R-transferService (5.15)	by UM	CREATE.inv
transfer-buffer-size	R-transferService (5.15)	by UM	CREATE.inv
latency-limit	R-transferService (5.15)	by UM	CREATE.inv
max-number-concatenations	R-transferService (5.15)	by UM	CREATE.inv
current-ts-version	TransferService (5.20)	by UM	CREATE.inv
lower-bound-reporting-period	TransferService (5.20)	by UM	CREATE.inv
time-out-period	TransferService (5.20)	by UM	CREATE.inv
transfer-service-state	TransferService (5.20)	= *waiting*	no
association-initiation-responsibility	TransferService (5.20)	by UM	CREATE.inv
responder-port-identifier	TransferService (5.20)	by UM	CREATE.inv
local-sle-entity-identifier	TransferService (5.20)	by CM	CREATE.inv
peer-sle-entity-access-list	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-start-time	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-stop-time	TransferService (5.20)	by UM	CREATE.inv
first-data-unit-time	TransferService (5.20)	by CM	when first data unit is transferred
last-data-unit-time	TransferService (5.20)	by CM	when last data unit is transferred
service-instance-start	TransferService (5.20)	= 'null'	when first service instance starts
service-instance-stop	TransferService (5.20)	= 'null'	when last service instance stops
provision-time	TransferService (5.20)	= 0	as needed
failed-binds	TransferService (5.20)	= 0	as needed
unavailable-time	TransferService (5.20)	= 0	as needed
done-time	TransferService (5.20)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

8.4.6 ACTIONS—R-cf-ts-p MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the R-transferService managed object class.

8.4.7 STATE-RELATED BEHAVIOR—R-cf-ts-p MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the R-transferService managed object class.

8.4.8 NOTIFICATIONS—R-cf-ts-p MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the R-transferService managed object class.

8.4.9 VALIDATION ASPECTS—R-cf-ts-p MO CLASS

a) The name of the channel provided shall be appropriate for an instance of return channel frames service.

b) The *allowed-channel-frame-dns* must exist in the channel tree and must match the channel type that the present managed object can process. If the channel cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

8.5 R-fsh-prod MANAGED OBJECT CLASS

8.5.1 PURPOSE

This managed object embodies the management interface to processes that extract Frame Secondary Header fields from Packet Telemetry Transfer Frames (Version 1 frames) that contain FSHs. This production is executed as part of the Return Frame Processing functional group.

8.5.2 INHERITANCE—R-fsh-prod MO CLASS

This managed object class is derived from and inherits the properties of the Production managed object class.

8.5.3 OBJECTS CONTAINED—R-fsh-prod MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero, one, or more r-fsh-st managed objects. This class is specified in 8.6.

b) Zero, one, or more r-fsh-ts-p managed objects. This class is specified in 8.7.

8.5.4 STATES—R-fsh-prod MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the Production managed object class.

8.5.5 ATTRIBUTES—R-fsh-prod MO CLASS

This managed object class has the following specific attributes:

a) **r-fsh-prod-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **fsh-lengths.** A list of (GVCID, fsh-length) pairs, with one entry in the list for each frame channel.

   1) Data type/value: List.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **Attribute Value Summary**. Refer to table 8-5.

Table 8-5: Attribute Value Summary—R-fsh-prod Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-fsh-prod-mo-id	—	by CM	CREATE.inv
fsh-lengths	—	by UM	CREATE.inv
production-state	Production (5.10)	= *waiting*	—
data-units-consumed-for-processing	Production (5.10)	= 0	as needed
processed-successfully	Production (5.10)	= 0	as needed
data-units-produced	Production (5.10)	= 0	as needed
number-invocations	Production (5.10)	= 0	as needed
unavailable-time	Production (5.10)	= 0	as needed
operational-time	Production (5.10)	= 0	as needed
production-start-time	Production (5.10)	by CM	when production starts
production-stop-time	Production (5.10)	by CM	when production stops
done-time	Production (5.10)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

8.5.6 ACTIONS—R-fsh-prod MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the Production managed object class.

8.5.7 STATE-RELATED BEHAVIOR—R-fsh-prod MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the Production managed object class.

8.5.8 NOTIFICATIONS—R-fsh-prod MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the Production managed object class.

8.5.9 VALIDATION ASPECTS—R-fsh-prod MO CLASS

No specific validation aspects are defined for this managed object class. However, validation aspects are inherited from the Production managed object class.

8.6 R-fsh-st MANAGED OBJECT CLASS

8.6.1 PURPOSE

This managed object represents the resources needed to store a return FSH channel. Refer to 5.3 for general storage considerations.

8.6.2 INHERITANCE—R-fsh-st MO CLASS

This managed object class is derived from and inherits the properties of the ChannelStored managed object class.

8.6.3 OBJECTS CONTAINED—R-fsh-st MO CLASS

An instance of this concrete managed object class shall contain zero, one, or more r-fsh-ts-p managed objects. Absence of an instance indicates that the data are only stored in the current service package. They may be retrieved in another service package. This class is specified in 8.7.

8.6.4 STATES—R-fsh-st MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the ChannelStored managed object class.

8.6.5 ATTRIBUTES—R-fsh-st MO CLASS

This managed object class has the following specific attributes:

a) **r-fsh-st-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

    1) Data type/value: Character string with a maximum length of 256 ASCII characters.

2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **stored-channel-frame-dns.** The set of frame channels that are stored by the storage resource represented by this managed object. These names must be formed according to the rules defined in annex C.

1) Data type/values: Set of DNs.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **Attribute Value Summary**. Refer to table 8-6.

Table 8-6: Attribute Value Summary—R-fsh-st Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-fsh-st-mo-id	—	by CM	CREATE.inv
stored-channel-frame-dns	—	by UM	CREATE.inv
data-store-id	ChannelStored (5.3)	by UM	CREATE.inv
partition-id	ChannelStored (5.3)	by UM	CREATE.inv
access-mode	ChannelStored (5.3)	by UM	CREATE.inv
ingest-window-start-time	ChannelStored (5.3)	by UM	CREATE.inv
ingest-window-stop-time	ChannelStored (5.3)	by UM	CREATE.inv
ingest-time	ChannelStored (5.3)	=0	CREATE.rtn
retrieval-window-start-time	ChannelStored (5.3)	by UM	CREATE.inv
retrieval-window-stop-time	ChannelStored (5.3)	by UM	CREATE.inv
unavailable-time	ChannelStored (5.3)	= 0	CREATE.rtn
provision-time	ChannelStored (5.3)	= 0	CREATE.rtn
number-SLE-PDUs-stored	ChannelStored (5.3)	= 0	CREATE.rtn
channel-stored-state	ChannelStored (5.3)	= *waiting*	CREATE.inv
done-time	ChannelStored (5.3)	by CM	when MO enters *done*
partition-data-units	ChannelStored (5.3)	= 0	as needed
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

8.6.6 ACTIONS—R-fsh-st MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the ChannelStored managed object class.

8.6.7 STATE-RELATED BEHAVIOR—R-fsh-st MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the ChannelStored managed object class.

8.6.8 NOTIFICATIONS—R-fsh-st MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the ChannelStored managed object class.

8.6.9 VALIDATION ASPECTS—R-fsh-st MO CLASS

a) The name of the channel stored shall be appropriate for an instance of return frame secondary header service.

b) The *stored-channel-frame-dns* must exist in the channel tree and must match the channel type that the present managed object can process. If the channels cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

8.7 R-fsh-ts-p MANAGED OBJECT CLASS

8.7.1 PURPOSE

This managed object represents the resources needed to provide a Return FSH transfer service. Refer to 5.20 for a general discussion of transfer service provision.

8.7.2 INHERITANCE—R-fsh-ts-p MO CLASS

This managed object class is derived from and inherits the properties of the R-transferService managed object class.

8.7.3 OBJECTS CONTAINED—R-fsh-ts-p MO CLASS

An instance of this concrete managed object class contains no other managed objects.

8.7.4 STATES—R-fsh-ts-p MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the R-transferService managed object class.

8.7.5 ATTRIBUTES—R-fsh-ts-p MO CLASS

This managed object class has the following specific attributes:

a) **r-fsh-ts-p-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string with a maximum length of 256 ASCII characters.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **allowed-fsh-channel-dns**. The set of DNs of the FSH data channels that may be selected by the user for transfer via the associated R-fsh transfer service.

   1) Data type/values: Set of DNs. Each DN must be formed according to the rules defined in annex C.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **FSH-data-units-sent**. The number of FSH data units sent.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

d) **FSH-data-units-supplied**. The number of FSH channel data units supplied by the service instance.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

e) **FSH-data-units-available**. The number of FSH channel data units available, but not supplied by the service instance, due to timely delivery requirements.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

f) **FSH-PDUs-supplied.** The number of FSH-PDUs supplied by the service instance.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

g) **Attribute Value Summary**. Refer to table 8-7.

Table 8-7: Attribute Value Summary—R-fsh-ts-p Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-fsh-ts-p-mo-id	—	by CM	CREATE.inv
allowed-fsh-channel-dns	—	by UM	CREATE.inv
fsh-data-units-sent	—	= 0	as needed
fsh-data-units-supplied	—	= 0	as needed
fsh-data-units-available	—	= 0	as needed
fsh-pdus-supplied	—	= 0	as needed
r-delivery-mode	R-transferService (5.15)	by UM	CREATE.inv
delivery-quality	R-transferService (5.15)	by UM	CREATE.inv
transfer-buffer-size	R-transferService (5.15)	by UM	CREATE.inv
latency-limit	R-transferService (5.15)	by UM	CREATE.inv
max-number-concatenations	R-transferService (5.15)	by UM	CREATE.inv
current-ts-version	TransferService (5.20)	by UM	CREATE.inv
lower-bound-reporting-period	TransferService (5.20)	by UM	CREATE.inv
time-out-period	TransferService (5.20)	by UM	CREATE.inv
transfer-service-state	TransferService (5.20)	= *waiting*	no
association-initiation-responsibility	TransferService (5.20)	by UM	CREATE.inv
responder-port-identifier	TransferService (5.20)	by UM	CREATE.inv
local-sle-entity-identifier	TransferService (5.20)	by CM	CREATE.inv
peer-sle-entity-access-list	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-start-time	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-stop-time	TransferService (5.20)	by UM	CREATE.inv
first-data-unit-time	TransferService (5.20)	by CM	when first data unit is transferred
last-data-unit-time	TransferService (5.20)	by CM	when last data unit is transferred
service-instance-start	TransferService (5.20)	= 'null'	when first service instance starts
service-instance-stop	TransferService (5.20)	= 'null'	when last service instance stops
provision-time	TransferService (5.20)	= 0	as needed
failed-binds	TransferService (5.20)	= 0	as needed
unavailable-time	TransferService (5.20)	= 0	as needed
done-time	TransferService (5.20)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

8.7.6 ACTIONS—R-fsh-ts-p MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the R-transferService managed object class.

8.7.7 STATE-RELATED BEHAVIOR—R-fsh-ts-p MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the R-transferService managed object class.

8.7.8 NOTIFICATIONS—R-fsh-ts-p MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the R-transferService managed object class.

8.7.9 VALIDATION ASPECTS—R-fsh-ts-p MO CLASS

a) The name of the channel provided shall be appropriate for an instance of return frame secondary header service.

b) The *allowed-fsh-channel-dns* must exist in the channel tree and must match the channel type that the present managed object can process. If the channels cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

8.8 R-ocf-prod MANAGED OBJECT CLASS

8.8.1 PURPOSE

This managed object embodies the management interface to the processes that extract Operational Control Fields (OCF) from Packet Telemetry Transfer Frames (Version 1 frames) or Virtual Channel Data Units (Version 2 frames) that contain OCFs. This production is executed as part of the Return Frame Processing functional group.

8.8.2 INHERITANCE—R-ocf-prod MO CLASS

This managed object class is derived from and inherits the properties of the Production managed object class.

8.8.3 OBJECTS CONTAINED—R-ocf-prod MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero, one, or more r-ocf-st managed objects. This class is specified in 8.9.

b) Zero, one, or more r-ocf-ts-p managed objects. This class is specified in 8.10.

8.8.4 STATES—R-ocf-prod MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the Production managed object class.

8.8.5 ATTRIBUTES—R-ocf-prod MO CLASS

This managed object class has the following specific attribute:

a) **r-ocf-prod-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **ocf-carrier-channels**. The *ocf-carrier-channels* is a list of GVCIDs, with one entry in the list for each frame channel containing an OCF.

   1) Data type/value: List of GVCIDs.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **Attribute Value Summary**. Refer to table 8-8.

Table 8-8: Attribute Value Summary—R-ocf-prod Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-ocf-prod-mo-id	—	by CM	CREATE.inv
ocf-carrier-channels	—	by UM	CREATE.inv
production-state	Production (5.10)	= *waiting*	—
data-units-consumed-for-processing	Production (5.10)	= 0	as needed
processed-successfully	Production (5.10)	= 0	as needed
data-units-produced	Production (5.10)	= 0	as needed
number-invocations	Production (5.10)	= 0	as needed
unavailable-time	Production (5.10)	= 0	as needed
operational-time	Production (5.10)	= 0	as needed
production-start-time	Production (5.10)	by CM	when production starts
production-stop-time	Production (5.10)	by CM	when production stops
done-time	Production (5.10)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

8.8.6 ACTIONS—R-ocf-prod MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the Production managed object class.

8.8.7 STATE-RELATED BEHAVIOR—R-ocf-prod MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the Production managed object class.

8.8.8 NOTIFICATIONS—R-ocf-prod MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the Production managed object class.

8.8.9 VALIDATION ASPECTS—R-ocf-prod MO CLASS

The name of the channel produced shall be appropriate to the service instance under consideration.

8.9 R-ocf-st MANAGED OBJECT CLASS

8.9.1 PURPOSE

This managed object represents the resources needed to store a return OCF channel. Refer to 5.3 for general storage considerations.

8.9.2 INHERITANCE—R-ocf-st MO CLASS

This managed object class is derived from and inherits the properties of the ChannelStored managed object class.

8.9.3 OBJECTS CONTAINED—R-ocf-st MO CLASS

An instance of this concrete managed object class shall contain zero, one, or more r-ocf-ts-p managed objects. Absence of an instance indicates that the data are only stored in the current service package. They may be retrieved in another service package. This class is specified in 8.10.

8.9.4 STATES—R-ocf-st MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the ChannelStored managed object class.

8.9.5 ATTRIBUTES—R-ocf-st MO CLASS

This managed object class has the following specific attributes:

a) **r-ocf-st-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/values: Character string with a maximum length of 256 ASCII characters.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **stored-channel-frame-dns.** The frame channels that are stored by the storage resource represented by this managed object.

   1) Data type/values: Set of DNs. These names must be formed according to the rules defined in annex C.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **Attribute Value Summary**. Refer to table 8-9.

Table 8-9: Attribute Value Summary—R-ocf-st Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-ocf-st-mo-id	—	by CM	CREATE.inv
stored-channel-frame-dns	—	by UM	CREATE.inv
data-store-id	ChannelStored (5.3)	by UM	CREATE.inv
partition-id	ChannelStored (5.3)	by UM	CREATE.inv
access-mode	ChannelStored (5.3)	by UM	CREATE.inv
ingest-window-start-time	ChannelStored (5.3)	by UM	CREATE.inv
ingest-window-stop-time	ChannelStored (5.3)	by UM	CREATE.inv
ingest-time	ChannelStored (5.3)	=0	CREATE.rtn
retrieval-window-start-time	ChannelStored (5.3)	by UM	CREATE.inv
retrieval-window-stop-time	ChannelStored (5.3)	by UM	CREATE.inv
unavailable-time	ChannelStored (5.3)	= 0	CREATE.rtn
provision-time	ChannelStored (5.3)	= 0	CREATE.rtn
number-SLE-PDUs-stored	ChannelStored (5.3)	= 0	CREATE.rtn
channel-stored-state	ChannelStored (5.3)	= *waiting*	CREATE.inv
done-time	ChannelStored (5.3)	by CM	when MO enters *done*
partition-data-units	ChannelStored (5.3)	= 0	as needed
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

8.9.6 ACTIONS—R-ocf-st MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the ChannelStored managed object class.

8.9.7 STATE-RELATED BEHAVIOR—R-ocf-st MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the ChannelStored managed object class.

8.9.8 NOTIFICATIONS—R-ocf-st MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the ChannelStored managed object class.

8.9.9 VALIDATION ASPECTS—R-ocf-st MO CLASS

a) The name of the channel stored shall be appropriate for an instance of return operational control field service.

b) The *stored-channel-frame-dns* must exist in the channel tree and must match the channel type that the present managed object can process. If the channels cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

8.10 R-ocf-ts-p MANAGED OBJECT CLASS

8.10.1 PURPOSE

This managed object represents the resources needed to provide a return OCF transfer service. Refer to 5.20 for a general discussion of transfer service provision.

8.10.2 INHERITANCE—R-ocf-ts-p MO CLASS

This managed object class is derived from and inherits the properties of the R-ocf-ts managed object class.

8.10.3 OBJECTS CONTAINED—R-ocf-ts-p MO CLASS

An instance of this concrete managed object class contains no other managed objects.

8.10.4 STATES—R-ocf-ts-p MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the R-ocf-ts managed object class.

8.10.5 ATTRIBUTES—R-ocf-ts-p MO CLASS

This managed object class has the following specific attributes:

a) **r-ocf-ts-p-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string with a maximum length of 256 ASCII characters.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **OCF-data-units-sent**. The number of operational control field (OCF) data units sent per frame channel.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

c) **OCF-data-units-supplied**. The number of OCF channel data units supplied by the service instance.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

d) **OCF-data-units-available**. The number of OCF channel data units available, but not supplied by the service instance, due to timely delivery requirements.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

e) **OCF-PDUs-supplied**. The number of OCF-PDUs supplied by the service instance.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

f) **ocf-transmission-period**. This attribute specifies the periodicity (in milliseconds) with which OCFs are to be transmitted when the *current-ocf-update-mode* = 'time-based'.

1) Data type/value: Integer in the range [1..10000], where 10000 corresponds to 10 seconds.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

g) **Attribute Value Summary**. Refer to table 8-10.

Table 8-10: Attribute Value Summary—R-ocf-ts-p Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-ocf-ts-p-mo-id	—	by CM	CREATE.inv
ocf-data-units-sent	—	= 0	as needed
ocf-data-units-supplied	—	= 0	as needed
ocf-data-units-available	—	= 0	as needed
ocf-pdus-supplied	—	= 0	as needed
ocf-transmission-period	—	by UM	CREATE.inv
ocf-type	R-ocf-ts (5.13)	by UM	CREATE.inv
allowed-ocf-update-mode	R-ocf-ts (5.13)	by UM	CREATE.inv
current-ocf-update-mode	R-ocf-ts (5.13)	='undefined'	No
allowed-tc-vc	R-ocf-ts (5.13)	by UM	CREATE.inv
current-tc-vc	R-ocf-ts (5.13)	='undefined'	No
allowed-ocf-channel-dns	R-ocf-ts (5.13)	by UM	CREATE.inv
r-delivery-mode	R-transferService (5.15)	by UM	CREATE.inv
delivery-quality	R-transferService (5.15)	by UM	CREATE.inv
transfer-buffer-size	R-transferService (5.15)	by UM	CREATE.inv
latency-limit	R-transferService (5.15)	by UM	CREATE.inv
max-number-concatenations	R-transferService (5.15)	by UM	CREATE.inv
current-ts-version	TransferService (5.20)	by UM	CREATE.inv
lower-bound-reporting-period	TransferService (5.20)	by UM	CREATE.inv
time-out-period	TransferService (5.20)	by UM	CREATE.inv
transfer-service-state	TransferService (5.20)	= *waiting*	no
association-initiation-responsibility	TransferService (5.20)	by UM	CREATE.inv
responder-port-identifier	TransferService (5.20)	by UM	CREATE.inv
local-sle-entity-identifier	TransferService (5.20)	by CM	CREATE.inv
peer-sle-entity-access-list	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-start-time	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-stop-time	TransferService (5.20)	by UM	CREATE.inv
first-data-unit-time	TransferService (5.20)	by CM	when first data unit is transferred
last-data-unit-time	TransferService (5.20)	by CM	when last data unit is transferred
service-instance-start	TransferService (5.20)	= 'null'	when first service instance starts
service-instance-stop	TransferService (5.20)	= 'null'	when last service instance stops
provision-time	TransferService (5.20)	= 0	as needed
failed-binds	TransferService (5.20)	= 0	as needed
unavailable-time	TransferService (5.20)	= 0	as needed
done-time	TransferService (5.20)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

8.10.6 ACTIONS—R-ocf-ts-p MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the R-ocf-ts managed object class.

8.10.7 STATE-RELATED BEHAVIOR—R-ocf-ts-p MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the R-ocf-ts managed object class.

8.10.8 NOTIFICATIONS—R-ocf-ts-p MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the R-ocf-ts managed object class.

8.10.9 VALIDATION ASPECTS—R-ocf-ts-p MO CLASS

a) The name of the channel provided shall be appropriate for an instance of return operational control field service.

b) The *allowed-sle-channel-dn* must exist in the channel tree and must match the channel type that the present managed object can process. If the channel cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

9 RETURN FRAME DATA EXTRACTION MANAGED OBJECT DEFINITIONS

9.1 R-bitstream-st MANAGED OBJECT CLASS

NOTE – This managed object supports the provision of r-bitstream services, which at the time of approval of this draft Recommendation have been identified in *the Cross Support Reference Model* (reference [1]), but has not yet been specified. Therefore, this managed object definition is currently for informative purposes only. When the service specification is developed and approved, this managed object definition will be updated as necessary to correspond to the as-specified SLE service, after which this managed object definition will become normative.

9.1.1 PURPOSE

This managed object represents the resources needed to store a return bitstream channel. Refer to 5.3 for general storage considerations.

9.1.2 INHERITANCE—R-bitstream-st MO CLASS

This managed object class is derived from and inherits the properties of the ChannelStored managed object class.

9.2 R-bitstream-ts-p MANAGED OBJECT CLASS

NOTE – This managed object supports the provision of r-bitstream services, which at the time of approval of this draft Recommendation have been identified in the *Cross Support Reference Model* (reference [1]), but has not yet been specified. Therefore, this managed object definition is currently for informative purposes only. When the service specification is developed and approved, this managed object definition will be updated as necessary to correspond to the as-specified SLE service, after which this managed object definition will become normative.

9.2.1 PURPOSE

This managed object represents the resources needed to provide a return bitstream transfer service. Refer to 5.20 for a general discussion of transfer service provision.

9.2.2 INHERITANCE—R-bitstream-ts-p MO CLASS

This managed object class is derived from and inherits the properties of the R-transferService managed object class.

9.3 R-cf-ts-u MANAGED OBJECT CLASS

9.3.1 PURPOSE

This managed object represents the resources needed to use a Return Frame transfer service. Refer to 5.20 for a general discussion of transfer service use.

9.3.2 INHERITANCE—R-cf-ts-u MO CLASS

This managed object class is derived from and inherits the properties of the R-transferService managed object class.

9.3.3 OBJECTS CONTAINED—R-cf-ts-u MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero or one r-frameDataField-prod managed object. This class is specified in 9.4.

b) Zero, one or more r-cf-st managed objects. This class is specified in 8.3.

9.3.4 STATES—R-cf-ts-u MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the R-transferService managed object class.

9.3.5 ATTRIBUTES—R-cf-ts-u MO CLASS

This managed object class has the following specific attributes:

a) **r-cf-ts-u-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

    1) Data type/value: Character string with a maximum length of 256 ASCII characters.

    2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **allowed-channel-frame-dns.** The SLE data channels that may be handled by the SLE resource represented by this managed object during the service instance life time.

    1) Data type/values: Set of DNs. These names must be formed according to the rules defined in annex C.

    2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **service-instance-identifier.** The service instance to be used.

1) Data type/value: DN. The value is equal to the DN of the transfer "service provided" managed object within the service-providing SLE Complex associated with the desired service instance.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

d) **RCF-data-units-consumed**. The number of RCF data units consumed.

1) Data type/value: Unsigned integer. Initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

e) **RCF-PDUs-consumed**. The number of RCF-PDUs consumed by the service instance.

1) Data type/value: Unsigned integer. Initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

f) **ert-interval-begin**. The value to be placed in the start-time parameter of the START invocation.

1) Data type/value: CCSDS time code.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

3) Dependencies: This attribute is applicable only if the value of *r-delivery-mode* is 'return offline'.

g) **ert-interval-end**. The value to be placed in the stop-time parameter of the START invocation.

1) Data type/values: CCSDS time code.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting.*

3) Dependencies: This attribute is applicable only if the value of *r-delivery-mode* is 'return offline'.

h) **Attribute Value Summary**. Refer to table 9-1.

Table 9-1: Attribute Value Summary—R-cf-ts-u Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-cf-ts-u-mo-id	—	by CM	CREATE.inv
allowed-channel-frame-dns	—	by UM	CREATE.inv
service-instance-identifier	—	by UM	CREATE.inv
rcf-data-units-consumed	—	= 0	as needed
rcf-pdus-consumed	—	= 0	as needed
ert-interval-begin	—	by UM	CREATE.inv
ert-interval-end	—	by UM	CREATE.inv
r-delivery-mode	R-transferService (5.15)	by UM	CREATE.inv
delivery-quality	R-transferService (5.15)	by UM	CREATE.inv
transfer-buffer-size	R-transferService (5.15)	by UM	CREATE.inv
latency-limit	R-transferService (5.15)	by UM	CREATE.inv
max-number-concatenations	R-transferService (5.15)	by UM	CREATE.inv
current-ts-version	TransferService (5.20)	by UM	CREATE.inv
lower-bound-reporting-period	TransferService (5.20)	by UM	CREATE.inv
time-out-period	TransferService (5.20)	by UM	CREATE.inv
transfer-service-state	TransferService (5.20)	= *waiting*	no
association-initiation-responsibility	TransferService (5.20)	by UM	CREATE.inv
responder-port-identifier	TransferService (5.20)	by UM	CREATE.inv
local-sle-entity-identifier	TransferService (5.20)	by CM	CREATE.inv
peer-sle-entity-access-list	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-start-time	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-stop-time	TransferService (5.20)	by UM	CREATE.inv
first-data-unit-time	TransferService (5.20)	by CM	when first data unit is transferred
last-data-unit-time	TransferService (5.20)	by CM	when last data unit is transferred
service-instance-start	TransferService (5.20)	= 'null'	when first service instance starts
service-instance-stop	TransferService (5.20)	= 'null'	when last service instance stops
provision-time	TransferService (5.20)	= 0	as needed
failed-binds	TransferService (5.20)	= 0	as needed
unavailable-time	TransferService (5.20)	= 0	as needed
done-time	TransferService (5.20)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

9.3.6 ACTIONS—R-cf-ts-u MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the R-transferService managed object class.

9.3.7 STATE-RELATED BEHAVIOR—R-cf-ts-u MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the R-transferService managed object class.

9.3.8 NOTIFICATIONS—R-cf-ts-u MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the R-transferService managed object class.

9.3.9 VALIDATION ASPECTS—R-cf-ts-u MO CLASS

a) At least one managed object shall be contained. If not, a ‘consistency defect’ shall be added to the containing servicePackage managed object's set of *defect-records*.

b) The *allowed-channel-frame-dns* must exist in the channel tree and must match the channel type that the present managed object can process. If the respective channels cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

c) The name of the channel used shall be appropriate for an instance of return channel frames service.

9.4 R-frameDataField-prod MANAGED OBJECT CLASS

9.4.1 PURPOSE

The purpose of the Return Frame Data Field managed object is to control and monitor the extraction of the Transfer Frame Data Field from Packet Telemetry Transfer Frames (PTTF) (Version 1 frames) and the VCDU Data Field from VCDUs (Version 2 [AOS] frames). This production is executed as part of the Return Frame Data Extraction functional group. This managed object specifies the leading and trailing offsets in the frames to find the contained frame data fields:

a) The leading offset shall be determined by whether there is an FSH (PTTF) or Insert (AOS) field, and the length of that field. Although the PTTF header signals both the presence and length of the FSH, *Packet Telemetry* (reference [2]) requires it to be a managed parameter. The presence and length of the insert field is not signaled in the AOS frame and must therefore always be managed.

b) The trailing offset is determined by whether there are OCF and/or frame error correction fields at the end of the frame. Although presence of the OCF is signaled in PTTF, *Packet Telemetry* (reference [2]) requires it to be a managed parameter. The presence of the OCF is not signaled in the AOS frame and must therefore always be managed. The presence of the frame error correction field is signaled in neither the PTTF nor AOS frame and therefore must always be managed.

9.4.2 INHERITANCE—R-frameDataField-prod MO CLASS

This managed object class is derived from and inherits the properties of the Production managed object class.

9.4.3 OBJECTS CONTAINED—R-frameDataField-prod MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero or one r-bitstream-st managed object. This class is specified in 9.1.

b) Zero, one, or more r-bitstream-ts-p managed object. This class is specified in 9.2.

c) Zero or one r-sp-st managed object. This class is specified in 9.5.

d) Zero, one, or more r-sp-ts-p managed object. This class is specified in 9.6.

9.4.4 STATES—R-frameDataField-prod MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the Production managed object class.

9.4.5 ATTRIBUTES—R-frameDataField-prod MO CLASS

This managed object class has the following specific attributes:

a) **r-frame-data-field-prod-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **r-fecf-flag.** The presence or absence of the CRC-based Frame Error Control Field (FECF) at the end of every frame processed by this managed object.

   1) Data type/values:

      i) 'fecf present'.

      ii) 'fecf not present'.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **ocf-flags.** The channels that contain OCFs.

1) Data type/values: List of GVCIDs, with one entry in the list for each frame channel containing an OCF.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

d) **fsh-lengths.** The length of the FSH in each frame channel that contains an FSH.

1) Data type/values: A list of (GVCID, fsh-length) pairs, with one entry in the list for each frame channel containing an FSH.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

e) **insert-length.** The length of the insert field in every frame processed by this managed object.

1) Data type/value: Unsigned integer.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

3) Dependencies: When no insert field is present, the value shall be set to zero. This attribute has a non-zero value only when all frames on the return physical channel are VCDUs (version 2 (AOS) frames), and when Insert Zones are present in the VCDUs.

f) **vcdu-header-error-control-used**. Specifies whether the VCDUs of the VC have the optional 2-octet VCDU Header Error Control field.

1) Data type/values:

i) 'not applicable'. The VC is a version 1 VC.

ii) 'used'. The VC is a version 2 VC and the error control field is used.

iii) 'not-used'. The VC is a version 2 VC and the error control field is not used.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

g) **Attribute Value Summary**. Refer to table 9-2.

Table 9-2: Attribute Value Summary—R-frameDataField-prod Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-frame-data-field-prod-mo-id	—	by CM	CREATE.inv
r-fecf-flag	—	by UM	CREATE.inv
ocf-flags	—	by UM	CREATE.inv
fsh-lengths	—	by UM	CREATE.inv
insert-length	—	by UM	CREATE.inv
vcdu-header-error-control-used	—	by UM	CREATE.inv
production-state	Production (5.10)	= *waiting*	—
data-units-consumed-for-processing	Production (5.10)	= 0	as needed
processed-successfully	Production (5.10)	= 0	as needed
data-units-produced	Production (5.10)	= 0	as needed
number-invocations	Production (5.10)	= 0	as needed
unavailable-time	Production (5.10)	= 0	as needed
operational-time	Production (5.10)	= 0	as needed
production-start-time	Production (5.10)	by CM	when production starts
production-stop-time	Production (5.10)	by CM	when production stops
done-time	Production (5.10)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

9.4.6 ACTIONS—R-frameDataField-prod MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the Production managed object class.

9.4.7 STATE-RELATED BEHAVIOR—R-frameDataField-prod MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the Production managed object class.

9.4.8 NOTIFICATIONS—R-frameDataField-prod MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the Production managed object class.

9.4.9 VALIDATION ASPECTS—R-frameDataField-prod MO CLASS

a) If the r-frameDataField-prod managed object is contained by an r-af-prod managed object (albeit removed by several levels of containment), the value of *r-fecf-flag* attribute of the contained r-frameDataField-prod managed object must be consistent with the *r-fec* attribute of the containing r-af-prod managed object. If not, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

b) If the r-frameDataField-prod managed object is contained by an r-af-prod managed object (albeit removed by several levels of containment) that also contains an r-insert-prod managed object, the value of the *insert-length* attribute of the contained r-frameDataField-prod managed object must be consistent with the *insert-length* attribute of the contained r-insert-prod managed object. If not, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

9.5 R-sp-st MANAGED OBJECT CLASS

9.5.1 PURPOSE

This managed object represents the resources needed to store an RSP channel. Refer to 5.3 for general storage considerations.

9.5.2 INHERITANCE—R-sp-st MO CLASS

This managed object class is derived from and inherits the properties of the ChannelStored managed object class.

9.5.3 OBJECTS CONTAINED—R-sp-st MO CLASS

An instance of this concrete managed object class shall contain zero, one, or more r-sp-ts-p managed objects. Absence of an instance indicates that the data are only stored in the current service package. They may be retrieved in another service package. This class is specified in 9.6.

9.5.4 STATES—R-sp-st MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the ChannelStored managed object class.

9.5.5 ATTRIBUTES—R-sp-st MO CLASS

This managed object class has the following specific attributes:

a) **r-sp-st-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string with a maximum length of 256 ASCII characters.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **stored-space-packet-dns.** The space packet channels that are stored by the storage resource represented by this managed object.

   1) Data type/values: Set of DNs. These names must be formed according to the rules defined in annex C.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **Attribute Value Summary**. Refer to table 9-3.

Table 9-3: Attribute Value Summary—R-sp-st Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-sp-st-mo-id	—	by CM	CREATE.inv
stored-space-packet-dns	—	by UM	CREATE.inv
data-store-id	ChannelStored (5.3)	by UM	CREATE.inv
partition-id	ChannelStored (5.3)	by UM	CREATE.inv
access-mode	ChannelStored (5.3)	by UM	CREATE.inv
ingest-window-start-time	ChannelStored (5.3)	by UM	CREATE.inv
ingest-window-stop-time	ChannelStored (5.3)	by UM	CREATE.inv
ingest-time	ChannelStored (5.3)	= 0	CREATE.rtn
retrieval-window-start-time	ChannelStored (5.3)	by UM	CREATE.inv
retrieval-window-stop-time	ChannelStored (5.3)	by UM	CREATE.inv
unavailable-time	ChannelStored (5.3)	= 0	CREATE.rtn
provision-time	ChannelStored (5.3)	= 0	CREATE.rtn
number-SLE-PDUs-stored	ChannelStored (5.3)	= 0	CREATE.rtn
channel-stored-state	ChannelStored (5.3)	= *waiting*	CREATE.inv
done-time	ChannelStored (5.3)	by CM	when MO enters *done*
partition-data-units	ChannelStored (5.3)	= 0	as needed
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

9.5.6 ACTIONS—R-sp-st MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the ChannelStored managed object class.

9.5.7 STATE-RELATED BEHAVIOR—R-sp-st MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the ChannelStored managed object class.

9.5.8 NOTIFICATIONS—R-sp-st MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the ChannelStored managed object class.

9.5.9 VALIDATION ASPECTS—R-sp-st MO CLASS

a) The name of the channel stored shall be appropriate for an instance of return space packet service.

b) The *stored-space-packet-dns* must exist in the channel tree and must match the channel type that the present managed object can process. If the respective channels cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

9.6 R-sp-ts-p MANAGED OBJECT CLASS

9.6.1 PURPOSE

This managed object represents the resources needed to provide an RSP transfer service. Refer to 5.20 for a general discussion of transfer service provision.

9.6.2 INHERITANCE—R-sp-ts-p MO CLASS

This managed object class is derived from and inherits the properties of the R-transferService managed object class.

9.6.3 OBJECTS CONTAINED—R-sp-ts-p MO CLASS

An instance of this concrete managed object class contains no other managed objects.

9.6.4 STATES—R-sp-ts-p MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the R-transferService managed object class.

9.6.5 ATTRIBUTES—R-sp-ts-p MO CLASS

This managed object class has the following specific attributes:

a) **r-sp-ts-p-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string with a maximum length of 256 ASCII characters.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **allowed-space-packet-dns.** The space packet channels handled by the SLE resource represented by this managed object during the service instance lifetime.

   1) Data type/values: Set of DNs. These names must be formed according to the rules defined in annex C.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **space-packets-sent**. The number of space packets sent.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

d) **RSP-data-units-supplied**. The number of RSP channel data units supplied by the service instance.

1) Data type/value: Unsigned integer. Initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

e) **RSP-data-units-available**. The number of RSP channel data units available, but not supplied by the service instance, due to timely delivery requirements.

1) Data type/value: Unsigned integer. Initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

f) **RSP-PDUs-supplied**. The number of RSP-PDUs supplied by the service instance.

1) Data type/value: Unsigned integer. Initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

g) **Attribute Value Summary**. Refer to table 9-4.

Table 9-4: Attribute Value Summary—R-sp-ts-p Managed Object

Attribute	Inherited From	Initial Value	Specified In
r-sp-ts-p-mo-id	—	by CM	CREATE.inv
allowed-space-packet-dns	—	by UM	CREATE.inv
space-packets-sent	—	= 0	as needed
rsp-data-units-supplied	—	= 0	as needed
rsp-data-units-available	—	= 0	as needed
rsp-pdus-supplied	—	= 0	as needed
r-delivery-mode	R-transferService (5.15)	by UM	CREATE.inv
delivery-quality	R-transferService (5.15)	by UM	CREATE.inv
transfer-buffer-size	R-transferService (5.15)	by UM	CREATE.inv
latency-limit	R-transferService (5.15)	by UM	CREATE.inv
max-number-concatenations	R-transferService (5.15)	by UM	CREATE.inv
current-ts-version	TransferService (5.20)	by UM	CREATE.inv
lower-bound-reporting-period	TransferService (5.20)	by UM	CREATE.inv
time-out-period	TransferService (5.20)	by UM	CREATE.inv
transfer-service-state	TransferService (5.20)	= *waiting*	no
association-initiation-responsibility	TransferService (5.20)	by UM	CREATE.inv
responder-port-identifier	TransferService (5.20)	by UM	CREATE.inv
local-sle-entity-identifier	TransferService (5.20)	by CM	CREATE.inv
peer-sle-entity-access-list	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-start-time	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-stop-time	TransferService (5.20)	by UM	CREATE.inv
first-data-unit-time	TransferService (5.20)	by CM	when first data unit is transferred
last-data-unit-time	TransferService (5.20)	by CM	when last data unit is transferred
service-instance-start	TransferService (5.20)	= 'null'	when first service instance starts
service-instance-stop	TransferService (5.20)	= 'null'	when last service instance stops
provision-time	TransferService (5.20)	= 0	as needed
failed-binds	TransferService (5.20)	= 0	as needed
unavailable-time	TransferService (5.20)	= 0	as needed
done-time	TransferService (5.20)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

9.6.6 ACTIONS—R-sp-ts-p MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the R-transferService managed object class.

9.6.7 STATE-RELATED BEHAVIOR—R-sp-ts-p MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the R-transferService managed object class.

9.6.8 NOTIFICATIONS—R-sp-ts-p MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the R-transferService managed object class.

9.6.9 VALIDATION ASPECTS—R-sp-ts-p MO CLASS

a) The name of the channel provided shall be appropriate for an instance of return space packet service.

b) The *allowed-space-packet-dns* must exist in the channel tree and must match the channel type that the present managed object can process. If the respective channel cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

10 FORWARD TC SPACE LINK PROCESSING MANAGED OBJECT DEFINITIONS

10.1 F-cltu-st MANAGED OBJECT CLASS

10.1.1 PURPOSE

This managed object represents the resources needed to store a forward CLTU channel. Refer to 5.3 for general storage considerations.

10.1.2 INHERITANCE—F-cltu-st MO CLASS

This managed object class is derived from and inherits the properties of the ChannelStored managed object class.

10.2 F-cltu-ts-p MANAGED OBJECT CLASS

10.2.1 PURPOSE

This managed object represents the resources needed to provide a forward CLTU transfer service. Refer to 5.20 for a general discussion of transfer service provision.

10.2.2 INHERITANCE—F-cltu-ts-p MO CLASS

This managed object class is derived from and inherits the properties of the F-transferService managed object class.

10.2.3 OBJECTS CONTAINED—F-cltu-ts-p MO CLASS

An instance of this concrete managed object class contains no other managed objects.

10.2.4 STATES—F-cltu-ts-p MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the TransferService managed object class.

10.2.5 ATTRIBUTES—F-cltu-ts-p MO CLASS

This managed object class has the following specific attributes:

a) **f-cltu-ts-p-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string with a maximum length of 256 ASCII characters.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **allowed-sle-channel-dn.** The SLE data channel that is handled by the SLE resource represented by this managed object.

   1) Data type/value: DN. This name must be formed according to the rules defined in annex C.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **CLTUs-consumed**. The number of CLTUs consumed.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

d) **CLTU-PDUs-consumed**. The number of F-CLTU-PDUs consumed by the service instance.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

e) **errored-CLTU-PDUs**. The number of incomplete or errored F-CLTU-PDUs encountered.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

f) **Attribute Value Summary**. Refer to table 10-1.

Table 10-1: Attribute Value Summary—F-cltu-ts-p Managed Object

Attribute	Inherited From	Initial Value	Specified in
f-cltu-ts-p-mo-id	—	by CM	CREATE.inv
allowed-sle-channel-dn	—	by UM	CREATE.inv
cltus-consumed	—	= 0	as needed
cltu-pdus-consumed	—	= 0	as needed
errored-cltu-pdus	—	= 0	as needed
f-delivery-mode	F-transferService (5.8)	by UM	CREATE.inv
current-ts-version	TransferService (5.20)	by UM	CREATE.inv
lower-bound-reporting-period	TransferService (5.20)	by UM	CREATE.inv
time-out-period	TransferService (5.20)	by UM	CREATE.inv
transfer-service-state	TransferService (5.20)	= waiting	no
association-initiation-responsibility	TransferService (5.20)	by UM	CREATE.inv
responder-port-identifier	TransferService (5.20)	by UM	CREATE.inv
local-sle-entity-identifier	TransferService (5.20)	by CM	CREATE.inv
peer-sle-entity-access-list	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-start-time	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-stop-time	TransferService (5.20)	by UM	CREATE.inv
first-data-unit-time	TransferService (5.20)	by CM	when first data unit is transferred
last-data-unit-time	TransferService (5.20)	by CM	when last data unit is transferred
service-instance-start	TransferService (5.20)	= 'null'	when first service instance starts
service-instance-stop	TransferService (5.20)	= 'null'	when last service instance stops
provision-time	TransferService (5.20)	= 0	as needed
failed-binds	TransferService (5.20)	= 0	as needed
unavailable-time	TransferService (5.20)	= 0	as needed
done-time	TransferService (5.20)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

10.2.6 ACTIONS—F-cltu-ts-p MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the F-transferService managed object class.

10.2.7 STATE-RELATED BEHAVIOR—F-cltu-ts-p MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the F-transferService managed object class.

10.2.8 NOTIFICATIONS—F-cltu-ts-p MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the F-transferService managed object class.

10.2.9 VALIDATION ASPECTS—F-cltu-ts-p MO CLASS

a) The name of the channel provided shall be appropriate for an instance of forward CLTU service.

b) The *allowed-sle-channel-dn* must exist in the channel tree and must match the channel type that the present managed object can process. If the channel cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

10.3 F-tc-session-prod MANAGED OBJECT CLASS

10.3.1 PURPOSE

This managed object embodies the management interface to the process that consumes a CLTU channel and performing the appropriate PLOP to form a TC physical channel. This production is executed as part of the Forward TC Space Link Processing functional group.

10.3.2 INHERITANCE—F-tc-session-prod MO CLASS

This managed object class is derived from and inherits the properties of the Production managed object class.

10.3.3 OBJECTS CONTAINED—F-tc-session-prod MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero or one f-cltu-prod managed objects. This class is specified in 11.1.

b) Zero or one f-cltu-st managed objects. This class is specified in 10.1.

c) Zero or one f-cltu-ts-p managed objects. This class is specified in 10.2.

10.3.4 STATES—F-tc-session-prod MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the Production managed object class.

10.3.5 ATTRIBUTES—F-tc-session-prod MO CLASS

This managed object class has the following specific attributes:

a) **f-tc-session-prod-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **version-number-clcw.** The version number of the CLCW that the resource needs to read.

   1) Data type/value: Unsigned integer between zero and three.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **Attribute Value Summary**. Refer to table 10-2.

Table 10-2: Attribute Value Summary—F-tc-session-prod Managed Object

Attribute	Inherited From	Initial Value	Specified in
f-tc-session-prod-mo-id	—	by CM	CREATE.inv
version-number-clcw	—	by UM	CREATE.inv
production-state	Production (5.10)	= *waiting*	—
data-units-consumed-for-processing	Production (5.10)	= 0	as needed
processed-successfully	Production (5.10)	= 0	as needed
data-units-produced	Production (5.10)	= 0	as needed
number-invocations	Production (5.10)	= 0	as needed
unavailable-time	Production (5.10)	= 0	as needed
operational-time	Production (5.10)	= 0	as needed
production-start-time	Production (5.10)	by CM	when production starts
production-stop-time	Production (5.10)	by CM	when production stops
done-time	Production (5.10)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

10.3.6 ACTIONS—F-tc-session-prod MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the Production managed object class.

10.3.7 STATE-RELATED BEHAVIOR—F-tc-session-prod MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the Production managed object class.

10.3.8 NOTIFICATIONS—F-tc-session-prod MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the Production managed object class.

10.3.9 VALIDATION ASPECTS—F-tc-session-prod MO CLASS

a) If required, one r-ocf-ts-u managed object shall be contained.

b) Exactly one f-cltu-st, f-cltu-ts-p, or f-cltu-prod must be contained. If not, a ‘consistency defect’ shall be added to the containing servicePackage managed object's set of *defect-records*.

11 FORWARD CLTU GENERATION MANAGED OBJECT DEFINITIONS

11.1 F-cltu-prod MANAGED OBJECT CLASS

11.1.1 PURPOSE

This managed object embodies the management interface to the processes that consume one or more TC-Frame data channels and convert them into CLTUs. This production is executed as part of the Forward CLTU Generation functional group.

11.1.2 INHERITANCE—F-cltu-prod MO CLASS

This managed object class is derived from and inherits the properties of the Production managed object class.

11.1.3 OBJECTS CONTAINED—F-cltu-prod MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero or one f-tc-f-st managed object. This class is specified in 11.3.

b) Zero or one f-tc-f-ts-p managed object. This class is specified in 11.4.

c) Zero or one f-vc-multiplex-prod managed object. This class is specified in 12.7.

11.1.4 STATES—F-cltu-prod MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the Production managed object class.

11.1.5 ATTRIBUTES—F-cltu-prod MO CLASS

This managed object class has the following specific attributes:

a) **f-cltu-prod-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **cltu-randomization.** Randomization is/is not being used. Refer to *Telecommand Part 1* (reference [6]), for a discussion of randomization.

   1) Data type/values:

      i) 'true' if randomization is applied.

ii) 'false' if randomization is not applied.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **Attribute Value Summary**. Refer to table 11-1.

Table 11-1: Attribute Value Summary—F-cltu-prod Managed Object

Attribute	Inherited From	Initial Value	Specified in
f-cltu-prod-mo-id	—	by CM	CREATE.inv
cltu-randomization	—	by UM	CREATE.inv
production-state	Production (5.10)	= *waiting*	—
data-units-consumed-for-processing	Production (5.10)	= 0	as needed
processed-successfully	Production (5.10)	= 0	as needed
data-units-produced	Production (5.10)	= 0	as needed
number-invocations	Production (5.10)	= 0	as needed
unavailable-time	Production (5.10)	= 0	as needed
operational-time	Production (5.10)	= 0	as needed
production-start-time	Production (5.10)	by CM	when production starts
production-stop-time	Production (5.10)	by CM	when production stops
done-time	Production (5.10)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

11.1.6 ACTIONS—F-cltu-prod MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the Production managed object class.

11.1.7 STATE-RELATED BEHAVIOR—F-cltu-prod MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the Production managed object class.

11.1.8 NOTIFICATIONS—F-cltu-prod MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the Production managed object class.

11.1.9 VALIDATION ASPECTS—F-cltu-prod MO CLASS

No specific validation aspects are defined for this managed object class. However, validation aspects are inherited from the Production managed object class.

11.2 F-cltu-ts-u MANAGED OBJECT CLASS

11.2.1 PURPOSE

This managed object represents the resources needed to use a forward CLTU transfer service. Refer to 5.20 for a general discussion of transfer service use.

11.2.2 INHERITANCE—F-cltu-ts-u MO CLASS

This managed object class is derived from and inherits the properties of the F-transferService managed object class.

11.2.3 OBJECTS CONTAINED—F-cltu-ts-u MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero or one f-cltu-prod managed object. This class is specified in 11.1.

b) Zero or one f-cltu-st managed object. This class is specified in 10.1.

11.2.4 STATES—F-cltu-ts-u MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the F-transferService managed object class.

11.2.5 ATTRIBUTES—F-cltu-ts-u MO CLASS

This managed object class has the following specific attributes:

a) **f-cltu-ts-u-mo-id**. The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string with maximum length of 256 ASCII characters.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **allowed-sle-channel-dn**. The SLE data channel that is handled by the SLE resource represented by this managed object.

   1) Data type/value: DN. This name must be formed according to the rules defined in annex C.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **service-instance-identifier**. The DN of the service instance to be used.

   1) Data type/value: The value is equal to the DN of the transfer “service provided” managed object within the service-providing SLE Complex associated with the desired service instance.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

d) **CLTUs-sent**. The number of CLTUs sent.

   1) Data type/value: Unsigned integer.

   2) Modifications:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

e) **CLTU-data-units-supplied**. The number of F-CLTU channel data units supplied by the service instance.

   1) Data type/value: Unsigned integer.

   2) Modifications:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

f) **CLTU-PDUs-supplied**. The number of F-CLTU-PDUs supplied by the service instance.

   1) Data type/value: Unsigned integer.

   2) Modifications:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

g) **errored-CLTU-PDUs**. The number of incomplete or errored F-CLTU-PDUs encountered.

   1) Data type/value: Unsigned integer.

   2) Modifications:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

h) **Attribute Value Summary**. Refer to table 11-2.

Table 11-2: Attribute Value Summary—F-cltu-ts-u Managed Object

Attribute	Inherited From	Initial Value	Specified in
f-cltu-ts-u-mo-id	—	by CM	CREATE.inv
allowed-sle-channel-dn	—	by UM	CREATE.inv
service-instance-identifier	—	by UM	CREATE.inv
cltus-sent	—	= 0	as needed
cltu-data-units-supplied	—	= 0	as needed
cltu-pdus-supplied	—	= 0	as needed
errored-cltu-pdus	—	= 0	as needed
f-delivery-mode	F-transferService (5.8)	by UM	CREATE.inv
current-ts-version	TransferService (5.20)	by UM	CREATE.inv
lower-bound-reporting-period	TransferService (5.20)	by UM	CREATE.inv
time-out-period	TransferService (5.20)	by UM	CREATE.inv
transfer-service-state	TransferService (5.20)	= *waiting*	no
association-initiation-responsibility	TransferService (5.20)	by UM	CREATE.inv
responder-port-identifier	TransferService (5.20)	by UM	CREATE.inv
local-sle-entity-identifier	TransferService (5.20)	by CM	CREATE.inv
peer-sle-entity-access-list	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-start-time	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-stop-time	TransferService (5.20)	by UM	CREATE.inv
first-data-unit-time	TransferService (5.20)	by CM	when first data unit is transferred
last-data-unit-time	TransferService (5.20)	by CM	when last data unit is transferred
service-instance-start	TransferService (5.20)	= 'null'	when first service instance starts
service-instance-stop	TransferService (5.20)	= 'null'	when last service instance stops
provision-time	TransferService (5.20)	= 0	as needed
failed-binds	TransferService (5.20)	= 0	as needed
unavailable-time	TransferService (5.20)	= 0	as needed
done-time	TransferService (5.20)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

11.2.6 ACTIONS—F-cltu-ts-u MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the F-transferService managed object class.

11.2.7 STATE-RELATED BEHAVIOR—F-cltu-ts-u MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the F-transferService managed object class.

11.2.8 NOTIFICATIONS—F-cltu-ts-u MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the F-transferService managed object class.

11.2.9 VALIDATION ASPECTS—F-cltu-ts-u MO CLASS

a) Only one managed object shall be contained. If not, a ‘consistency defect’ shall be added to the containing servicePackage managed object's set of *defect-records*.

b) The name of the channel used shall be appropriate for an instance of forward CLTU service.

c) The *allowed-sle-channel-dn* must exist in the channel tree and must match the channel type that the present managed object can process. If the channel cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

11.3 F-tc-f-st MANAGED OBJECT CLASS

11.3.1 PURPOSE

This managed object represents the resources needed to store a Forward Telecommand Frames (FTCF) channel. Refer to 5.3 for general storage considerations.

11.3.2 INHERITANCE—F-tc-f-st MO CLASS

This managed object class is derived from and inherits the properties of the ChannelStored managed object class.

11.4 F-tc-f-ts-p MANAGED OBJECT CLASS

11.4.1 PURPOSE

This managed object represents the resources needed to provide a FTCF transfer service. Refer to 5.20 for a general discussion of transfer service provision.

11.4.2 INHERITANCE—F-tc-f-ts-p MO CLASS

This managed object class is derived from and inherits the properties of the F-transferService managed object class.

11.4.3 OBJECTS CONTAINED—F-tc-f-ts-p MO CLASS

An instance of this concrete managed object class contains no other managed objects.

11.4.4 STATES—F-tc-f-ts-p MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the F-transferService managed object class.

11.4.5 ATTRIBUTES—F-tc-f-ts-p MO CLASS

This managed object class has the following specific attributes:

a) **f-tc-f-ts-p-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string with a maximum length of 256 ASCII characters.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **allowed-sle-channel-dn.** The SLE data channel that is handled by the SLE resource represented by this managed object. This name must be formed according to the rules defined in annex C.

   1) Data type/value: DN.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **max-block-length.** The maximum length (in octets) of a block of TC frames.

   1) Data type/value: Unsigned integer.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

d) **data-units-consumed**. The number of data units consumed.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

e) **sequence-controlled-data-units**. The number of input data units confirmed as transferred via sequence-controlled mode.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

f) **last-reported-V(S)**. The last reported V(S) (next expected frame number onboard) from CLCW.

   1) Data type/value: Unsigned integer. Initial value shall be 'null'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

g) **FTCF-data-units-consumed**. The number of FTCF channel data units consumed by the service instance.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

h) **FTCF-PDUs-consumed**. The number of FTCF-PDUs consumed by the service instance.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

i) **errored-FTCF-PDUs**. The number of incomplete or errored FTCF-PDUs encountered.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase.

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

j) **Attribute Value Summary**. Refer to table 11-3.

Table 11-3: Attribute Value Summary—F-tc-f-ts-p Managed Object

Attribute	Inherited From	Initial Value	Specified in
f-tc-f-ts-p-mo-id	—	by CM	CREATE.inv
allowed-sle-channel-dn	—	by UM	CREATE.inv
max-block-length	—	by UM	CREATE.inv
data-units-consumed	—	= 0	as needed
sequence-controlled-data-units	—	= 0	as needed
last-reported-V(S)	—	by CM	as needed
ftcf-data-units-consumed	—	= 0	as needed
ftcf-pdus-consumed	—	= 0	as needed
errored-ftcf-pdus	—	= 0	as needed
f-delivery-mode	F-transferService (5.8)	by UM	CREATE.inv
current-ts-version	TransferService (5.20)	by UM	CREATE.inv
lower-bound-reporting-period	TransferService (5.20)	by UM	CREATE.inv
time-out-period	TransferService (5.20)	by UM	CREATE.inv
transfer-service-state	TransferService (5.20)	= *waiting*	no
association-initiation-responsibility	TransferService (5.20)	by UM	CREATE.inv
responder-port-identifier	TransferService (5.20)	by UM	CREATE.inv
local-sle-entity-identifier	TransferService (5.20)	by CM	CREATE.inv
peer-sle-entity-access-list	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-start-time	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-stop-time	TransferService (5.20)	by UM	CREATE.inv
first-data-unit-time	TransferService (5.20)	by CM	when first data unit is transferred
last-data-unit-time	TransferService (5.20)	by CM	when last data unit is transferred
service-instance-start	TransferService (5.20)	= 'null'	when first service instance starts
service-instance-stop	TransferService (5.20)	= 'null'	when last service instance stops
provision-time	TransferService (5.20)	= 0	as needed
failed-binds	TransferService (5.20)	= 0	as needed
unavailable-time	TransferService (5.20)	= 0	as needed
done-time	TransferService (5.20)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

11.4.6 ACTIONS—F-tc-f-ts-p MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the F-transferService managed object class.

11.4.7 STATE-RELATED BEHAVIOR—F-tc-f-ts-p MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the F-transferService managed object class.

11.4.8 NOTIFICATIONS—F-tc-f-ts-p MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the F-transferService managed object class.

11.4.9 VALIDATION ASPECTS—F-tc-f-ts-p MO CLASS

a) The name of the channel provided shall be appropriate for an instance of forward telecommand frames service.

b) The *allowed-sle-channel-dn* must exist in the channel tree and must match the channel type that the present managed object can process. If the channel cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

12 FORWARD TC VC DATA INSERTION MANAGED OBJECT DEFINITIONS

12.1 F-sp-st MANAGED OBJECT CLASS

12.1.1 PURPOSE

This managed object represents the resources needed to store a Forward Space Packet (FSP) channel. Refer to 5.3 for general storage considerations.

12.1.2 INHERITANCE—F-sp-st MO CLASS

This managed object class is derived from and inherits the properties of the ChannelStored managed object class.

12.2 F-sp-ts-p MANAGED OBJECT CLASS

12.2.1 PURPOSE

This managed object represents the resources needed to provide an FSP transfer service. Refer to 5.20 for a general discussion of transfer service provision.

12.2.2 INHERITANCE—F-sp-ts-p MO CLASS

This managed object class is derived from and inherits the properties of the F-transferService managed object class.

12.2.3 OBJECTS CONTAINED—F-sp-ts-p MO CLASS

An instance of this concrete managed object class contains no other managed objects.

12.2.4 STATES—F-sp-ts-p MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the F-transferService managed object class.

12.2.5 ATTRIBUTES—F-sp-ts-p MO CLASS

This managed object class has the following specific attributes:

a) **f-sp-ts-p-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string with a maximum length of 256 ASCII characters.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **allowed-space-packet-dns.** The set of space packet channels handled by the SLE resource represented by this managed object.

   1) Data type/values: Set of DNs. These names must be formed according to the rules defined in annex C.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **used-maps.** The set of the MAPs assigned for use by this service instance.

NOTE – Each MAP within a given VC can be used by only one FSP service instance.

   1) Data type/value: Set of unsigned integers between 0 and 63 (inclusive).

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

d) **FSPs-consumed**. The number of forward space packets consumed.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

e) **FSP-data-units-consumed**. The number of FSP channel data units consumed by the service instance.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

f) **FSP-PDUs-consumed**. The number of FSP-PDUs consumed by the service instance.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

g) **errored-FSP-PDUs**. The number of incomplete or errored FSP-PDUs encountered.

1) Data type/value: Unsigned integer. Initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

h) **allowed-transmission-mode.** The transmission mode of the packet.

1) Data type/values:

   i) 'sequence-controlled'. The service instance is allowed to send data only in the sequence controlled (AD) mode.

   ii) 'expedited'. The service instance is allowed to send data only in the expedited (BD) mode.

   iii) 'any'. The service instance is allowed to send data with or without sequence control.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

i) **Attribute Value Summary**. Refer to table 12-1.

Table 12-1: Attribute Value Summary—F-sp-ts-p Managed Object

Attribute	Inherited From	Initial Value	Specified in
f-sp-ts-p-mo-id	—	by CM	CREATE.inv
allowed-space-packet-dns	—	by UM	CREATE.inv
used-maps	—	by UM	CREATE.inv
fsps-consumed	—	= 0	as needed
fsp-data-units-consumed	—	= 0	as needed
fsp-pdus-consumed	—	= 0	as needed
errored-fsp-pdus	—	= 0	as needed
allowed-transmision-mode	—	by UM	CREATE.inv
f-delivery-mode	F-transferService (5.8)	by UM	CREATE.inv
current-ts-version	TransferService (5.20)	by UM	CREATE.inv
lower-bound-reporting-period	TransferService (5.20)	by UM	CREATE.inv
time-out-period	TransferService (5.20)	by UM	CREATE.inv
transfer-service-state	TransferService (5.20)	= *waiting*	no
association-initiation-responsibility	TransferService (5.20)	by UM	CREATE.inv
responder-port-identifier	TransferService (5.20)	by UM	CREATE.inv
local-sle-entity-identifier	TransferService (5.20)	by CM	CREATE.inv
peer-sle-entity-access-list	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-start-time	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-stop-time	TransferService (5.20)	by UM	CREATE.inv
first-data-unit-time	TransferService (5.20)	by CM	when first data unit is transferred
last-data-unit-time	TransferService (5.20)	by CM	when last data unit is transferred
service-instance-start	TransferService (5.20)	= 'null'	when first service instance starts
service-instance-stop	TransferService (5.20)	= 'null'	when last service instance stops
provision-time	TransferService (5.20)	= 0	as needed
failed-binds	TransferService (5.20)	= 0	as needed
unavailable-time	TransferService (5.20)	= 0	as needed
done-time	TransferService (5.20)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

12.2.6 ACTIONS—F-sp-ts-p MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the F-transferService managed object class.

12.2.7 STATE-RELATED BEHAVIOR—F-sp-ts-p MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the F-transferService managed object class.

12.2.8 NOTIFICATIONS—F-sp-ts-p MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the F-transferService managed object class.

12.2.9 VALIDATION ASPECTS—F-sp-ts-p MO CLASS

a) The name of the channel provided shall be appropriate for an instance of forward space packet service.

b) The *allowed-space-packet-dns* must exist in the channel tree and must match the channel type that the present managed object can process. If the channel cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

12.3 F-tc-f-ts-u MANAGED OBJECT CLASS

12.3.1 PURPOSE

This managed object represents the resources needed to use an FTCF transfer service. Refer to 5.20 for a general discussion of transfer service use.

12.3.2 INHERITANCE—F-tc-f-ts-u MO CLASS

This managed object class is derived from and inherits the properties of the F-transferService managed object class.

12.3.3 OBJECTS CONTAINED—F-tc-f-ts-u MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero or one f-vc-multiplex-prod managed object. This class is specified in 12.7.

b) Zero or one f-tc-f-st managed object. This class is specified in 11.3.

12.3.4 STATES—F-tc-f-ts-u MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the F-transferService managed object class.

12.3.5 ATTRIBUTES—F-tc-f-ts-u MO CLASS

This managed object class has the following specific attributes:

a) **f-tc-f-ts-u-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

1) Data type/value: Character string with a maximum length of 256 ASCII characters.

2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **allowed-sle-channel-dn.** The SLE data channel that is handled by the SLE resource represented by this managed object.

1) Data type/value: DN. This name must be formed according to the rules defined in annex C.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **max-block-length**. The maximum length (in octets) of a block of TC frames.

1) Data type/value: Unsigned integer.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

d) **service-instance-identifier.** The service instance to be used.

1) Data type/value: DN of the transfer "service provided" managed object within the service-providing SLE Complex associated with the desired service instance.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

e) **data-units-sent**. The number of data units sent.

1) Data type/value: Unsigned integer. Initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

f) **FTCF-data-units-supplied**. The number of FTCF channel data units supplied by the service instance.

1) Data type/value: Unsigned integer. Initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase:

i) SET shall be disabled in both the definition and utilization phases.

ii) GET shall be disabled in the definition phase.

g) **FTCF-PDUs-supplied**. The number of FTCF-PDUs supplied by the service instance.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

h) **errored-FTCF-PDUs**. The number of incomplete or errored FTCF-PDUs encountered.

   1) Data type/value: Unsigned integer. Initial value shall be '0'.

   2) Modifications: SLE Complex Management may modify the value during the utilization phase:

      i) SET shall be disabled in both the definition and utilization phases.

      ii) GET shall be disabled in the definition phase.

i) **Attribute Value Summary**. Refer to table 12-2.

Table 12-2: Attribute Value Summary—F-tc-f-ts-u Managed Object

Attribute	Inherited From	Initial Value	Specified in
f-tc-f-ts-u-mo-id	—	by CM	CREATE.inv
allowed-sle-channel-dn	—	by UM	CREATE.inv
max-block-length	—	by UM	CREATE.inv
service-instance-identifier	—	by UM	CREATE.inv
data-units-sent	—	= 0	as needed
ftcf-data-units-supplied	—	= 0	as needed
ftcf-pdus-supplied	—	= 0	as needed
errored-ftcf-pdus	—	= 0	as needed
f-delivery-mode	F-transferService (5.8)	by UM	CREATE.inv
current-ts-version	TransferService (5.20)	by UM	CREATE.inv
lower-bound-reporting-period	TransferService (5.20)	by UM	CREATE.inv
time-out-period	TransferService (5.20)	by UM	CREATE.inv
transfer-service-state	TransferService (5.20)	= *waiting*	no
association-initiation-responsibility	TransferService (5.20)	by UM	CREATE.inv
responder-port-identifier	TransferService (5.20)	by UM	CREATE.inv
local-sle-entity-identifier	TransferService (5.20)	by CM	CREATE.inv
peer-sle-entity-access-list	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-start-time	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-stop-time	TransferService (5.20)	by UM	CREATE.inv
first-data-unit-time	TransferService (5.20)	by CM	when first data unit is transferred
last-data-unit-time	TransferService (5.20)	by CM	when last data unit is transferred
service-instance-start	TransferService (5.20)	= 'null'	when first service instance starts
service-instance-stop	TransferService (5.20)	= 'null'	when last service instance stops
provision-time	TransferService (5.20)	= 0	as needed
failed-binds	TransferService (5.20)	= 0	as needed
unavailable-time	TransferService (5.20)	= 0	as needed
done-time	TransferService (5.20)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

12.3.6 ACTIONS—F-tc-f-ts-u MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the F-transferService managed object class.

12.3.7 STATE-RELATED BEHAVIOR—F-tc-f-ts-u MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the F-transferService managed object class.

12.3.8 NOTIFICATIONS—F-tc-f-ts-u MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the F-transferService managed object class.

12.3.9 VALIDATION ASPECTS—F-tc-f-ts-u MO CLASS

a) Only one managed object shall be contained. If not, a ‘consistency defect’ record shall be added to the containing managed object's set of *defect-records*.

b) The name of the channel used shall be appropriate for an instance of forward telecommand frames service.

c) The *allowed-sle-channel-dn* must exist in the channel tree and must match the channel type that the present managed object can process. If the channel cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

12.4 F-tc-vc-prod MANAGED OBJECT CLASS

12.4.1 PURPOSE

This managed object embodies the management interface to the functionality that constructs TC frames to carry TC Frame Data Units and executes appropriate aspects of the COP. This production is executed as part of the Forward TC-VC-Data Insertion functional group.

12.4.2 INHERITANCE—F-tc-vc-prod MO CLASS

This managed object class is derived from and inherits the properties of the Production managed object class.

12.4.3 OBJECTS CONTAINED—F-tc-vc-prod MO CLASS

An instance of this concrete managed object class shall contain <u>only one</u> instance of the following classes:

a) One f-vc-seg-prod managed object. An f-vc-seg-prod managed object is contained only if the VC carries CCSDS packets. This class is specified in 12.8.

b) One f-tc-vca-ts-p managed object. An f-tc-vca-ts-p managed object is contained only if the VC carries online TC-VCA data units. This class is specified in 12.6.

c) One f-tc-vca-st managed object. An f-tc-vca-st managed object is contained only if the VC carries offline TC-VCA data units. This class is specified in 12.5.

12.4.4 STATES—F-tc-vc-prod MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the Production managed object class.

12.4.5 ATTRIBUTES—F-tc-vc-prod MO CLASS

This managed object class has the following specific attributes:

a) **f-tc-vc-prod-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **vc-id.** The identifier of the virtual channel created by this managed object.

   1) Data type/value: integer between 0 and 63, inclusive.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

   NOTE – The value of *vc-id* is placed in the VIRTUAL CHANNEL ID field of Frame Headers of the TC Transfer Frames generated by the resources represented by this managed object. The value of the SPACECRAFT ID field is specified in the *sc-id* attribute of the f-vc-multiplex-prod managed object that contains the f-tc-vc-prod managed object.

c) **Attribute Value Summary**. Refer to table 12-3.

Table 12-3: Attribute Value Summary—F-tc-vc-prod Managed Object

Attribute	Inherited From	Initial Value	Specified in
f-tc-vc-prod-mo-id	—	by CM	CREATE.inv
vc-id	—	by UM	CREATE.inv
production-state	Production (5.10)	= *waiting*	—
data-units-consumed-for-processing	Production (5.10)	= 0	as needed
processed-successfully	Production (5.10)	= 0	as needed
data-units-produced	Production (5.10)	= 0	as needed
number-invocations	Production (5.10)	= 0	as needed
unavailable-time	Production (5.10)	= 0	as needed
operational-time	Production (5.10)	= 0	as needed
production-start-time	Production (5.10)	by CM	when production starts
production-stop-time	Production (5.10)	by CM	when production stops
done-time	Production (5.10)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

12.4.6 ACTIONS—F-tc-vc-prod MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the Production managed object class.

12.4.7 STATE-RELATED BEHAVIOR—F-tc-vc-prod MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the Production managed object class.

12.4.8 NOTIFICATIONS—F-tc-vc-prod MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the Production managed object class.

12.4.9 VALIDATION ASPECTS—F-tc-vc-prod MO CLASS

No specific validation aspects are defined for this managed object class. However, validation aspects are inherited from the Production managed object class.

12.5 F-tc-vca-st MANAGED OBJECT CLASS

12.5.1 PURPOSE

This managed object represents the resources needed to store a forward TC-VCA channel. Refer to 5.3 for general storage considerations.

12.5.2 INHERITANCE—F-tc-vca-st MO CLASS

This managed object class is derived from and inherits the properties of the ChannelStored managed object class.

12.6 F-tc-vca-ts-p MANAGED OBJECT CLASS

12.6.1 PURPOSE

This managed object represents the resources needed to provide a TC VC Access transfer service. Refer to 5.20 for a general discussion of transfer service provision.

12.6.2 INHERITANCE—F-tc-vca-ts-p MO CLASS

This managed object class is derived from and inherits the properties of the F-transferService managed object class.

12.6.3 OBJECTS CONTAINED—F-tc-vca-ts-p MO CLASS

An instance of this concrete managed object class contains no other managed objects.

12.6.4 STATES—F-tc-vca-ts-p MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the F-transferService managed object class.

12.6.5 ATTRIBUTES—F-tc-vca-ts-p MO CLASS

This managed object class has the following specific attributes:

a) **f-tc-vca-ts-p-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string with a maximum length of 256 ASCII characters.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **allowed-sle-channel-dn.** The SLE data channel that is handled by the SLE resource represented by this managed object.

   1) Data type/value: DN. This name must be formed according to the rules defined in annex C.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **TCVCA-data-units-consumed**. The number of forward telecommand virtual channel access data units consumed.

1) Data type/value: Unsigned integer. Initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

d) **TCVCA-PDUs-consumed.** The number of FTCVCA-PDUs consumed by the service instance.

1) Data type/value: Unsigned integer. Initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

e) **errored-TCVCA-PDUs**. The number of incomplete or errored FTCVCA-PDUs encountered.

1) Data type/value: Unsigned integer. Initial value shall be '0'.

2) Modifications: SLE Complex Management may modify the value during the utilization phase:

   i) SET shall be disabled in both the definition and utilization phases.

   ii) GET shall be disabled in the definition phase.

f) **allowed-transmission-mode.** The transmission mode of the packet.

1) Data type/values:

   i) 'sequence-controlled'. The service instance is allowed to send data only in the sequence controlled (AD) mode.

   ii) 'expedited'. The service instance is allowed to send data only in the expedited (BD) mode.

   iii) 'any'. The service instance is allowed to send data with or without sequence control.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

g) **Attribute Value Summary**. Refer to table 12-4.

Table 12-4: Attribute Value Summary—F-tc-vca-ts-p Managed Object

Attribute	Inherited From	Initial Value	Specified in
f-tc-vca-ts-p-mo-id	—	by CM	CREATE.inv
allowed-sle-channel-dn	—	by UM	CREATE.inv
tcvca-data-units-consumed	—	= 0	as needed
tcvca-pdus-consumed	—	= 0	as needed
errored-tcvca-pdus	—	= 0	as needed
allowed-transmission-mode	—	by UM	CREATE.inv
f-delivery-mode	F-transferService (5.8)	by UM	CREATE.inv
current-ts-version	TransferService (5.20)	by UM	CREATE.inv
lower-bound-reporting-period	TransferService (5.20)	by UM	CREATE.inv
time-out-period	TransferService (5.20)	by UM	CREATE.inv
transfer-service-state	TransferService (5.20)	= *waiting*	no
association-initiation-responsibility	TransferService (5.20)	by UM	CREATE.inv
responder-port-identifier	TransferService (5.20)	by UM	CREATE.inv
local-sle-entity-identifier	TransferService (5.20)	by CM	CREATE.inv
peer-sle-entity-access-list	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-start-time	TransferService (5.20)	by UM	CREATE.inv
scheduled-period-stop-time	TransferService (5.20)	by UM	CREATE.inv
first-data-unit-time	TransferService (5.20)	by CM	when first data unit is transferred
last-data-unit-time	TransferService (5.20)	by CM	when last data unit is transferred
service-instance-start	TransferService (5.20)	= 'null'	when first service instance starts
service-instance-stop	TransferService (5.20)	= 'null'	when last service instance stops
provision-time	TransferService (5.20)	= 0	as needed
failed-binds	TransferService (5.20)	= 0	as needed
unavailable-time	TransferService (5.20)	= 0	as needed
done-time	TransferService (5.20)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

12.6.6 ACTIONS—F-tc-vca-ts-p MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the F-transferService managed object class.

12.6.7 STATE-RELATED BEHAVIOR—F-tc-vca-ts-p MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the F-transferService managed object class.

12.6.8 NOTIFICATIONS—F-tc-vca-ts-p MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the F-transferService managed object class.

12.6.9 VALIDATION ASPECTS—F-tc-vca-ts-p MO CLASS

a) The name of the channel provided shall be appropriate for an instance of forward telecommand virtual channel access service.

b) The *allowed-sle-channel-dn* must exist in the channel tree and must match the channel type that the present managed object can process. If the channel cannot be found or if there is a channel type mismatch, a 'consistency defect' shall be added to the containing servicePackage managed object's set of *defect-records*.

12.7 F-vc-multiplex-prod MANAGED OBJECT CLASS

12.7.1 PURPOSE

This managed object embodies the management interface to the process that multiplexes multiple TC VCs onto a single F-TC-F data channel. This production is executed as part of the Forward TC-VC-Data Insertion functional group.

12.7.2 INHERITANCE—F-vc-multiplex-prod MO CLASS

This managed object class is derived from and inherits the properties of the Production managed object class.

12.7.3 OBJECTS CONTAINED—F-vc-multiplex-prod MO CLASS

An instance of this concrete managed object class shall contain one or more f-tc-vc-prod managed objects. One f-tc-vc-prod managed object must be contained for each forward TC-VC that is processed by the resource represented by this managed object. This class is specified in 12.4.

12.7.4 STATES—F-vc-multiplex-prod MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the Production managed object class.

12.7.5 ATTRIBUTES—F-vc-multiplex-prod MO CLASS

This managed object class has the following specific attributes:

a) **f-vc-multiplex-prod-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/value: Character string.

2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **max-frame-length.** The maximum TC transfer frame length (in number of octets).

1) Data type/value: Unsigned integer between five and 1024.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **tc-fecf-flag.** The TC transfer frame error control field must/must not be inserted in the transfer frame.

1) Data type/values:

i) 'true. if the TC frame error control field must be inserted.

ii) 'false'. if the TC frame error control field must not be inserted.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

d) **vc-multiplexing-scheme.** The VC multiplexing scheme.

1) Data type/values:

i) 'fifo'. The requests are served first-in-first-out.

ii) 'absolute-priority'. The highest priority request is always served first.

iii) 'polling-vector'. Requests are served according to a predefined polling vector.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

e) **max-bit-rate.** The maximum data transfer rate that the underlying transfer service supports (in bits/sec).

1) Data type/value: Unsigned integer.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

f) **vc-multiplexing-control.** The polling priority of the *vc-multiplexing-scheme.*

1) Data type/value: See item 3 below.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

3) Dependencies: The value of this attribute depends on the value of *vc-multiplexing-scheme* as follows:

i) If the value of *vc-multiplexing-scheme* is 'fifo', the value of *vc-multiplexing-control* shall be 'null'.

ii) If the value of *vc-multiplexing-scheme* is 'absolute priority', the value of *vc-multiplexing-control* shall be a sequence of up to 64 VCIDs, where VCID is an integer (value 0-63). The first value in the sequence shall be the VCID of the highest-priority VC, and the second value is the VCID of the second-highest priority, etc.

iii) If the value of *vc-multiplexing-scheme* is 'polling vector', the value of *vc-multiplexing-control* shall be a sequence of up to 192 VCIDs, where VCID is an integer (value 0-63). The first value in the sequence shall be the VCID of the VC to be polled first, the second value is the VCID of the VC to be polled second, etc.

NOTES

1) The f-vc-multiplex-prod managed object is used only for Telecommand, so it is implicit that only Version 1 (Telecommand) frames and SCIDs apply.

2) A VCID may appear more than once in the polling vector (e.g., in the case where a high-data-rate VC must be polled more often than other lower-rate VCs).

g) **sc-id.** The spacecraft identifier associated with the virtual channels that are multiplexed into the TC Frame channel produced by this managed object.

1) Data type/value: Integer between 0 and 1023, inclusive.

2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

NOTE – The value of *sc-id* is shared by the f-tc-vc-prod managed objects contained by this f-vc-multiplex-prod managed object. The value of *sc-id* is placed in the SPACECRAFT ID field of Frame Headers of the TC Transfer Frames generated by the resources represented by the f-tc-vc-prod managed objects.

h) **Attribute Value Summary**. Refer to table 12-5.

Table 12-5: Attribute Value Summary—F-vc-multiplex-prod Managed Object

Attribute	Inherited From	Initial Value	Specified in
f-vc-multiplex-prod-mo-id	—	by CM	CREATE.inv
max-frame-length	—	by UM	CREATE.inv
tc-fecf-flag	—	by UM	CREATE.inv
vc-multiplexing-scheme	—	by UM	CREATE.inv
max-bit-rate	—	by UM	CREATE.inv
vc-multiplexing-control	—	by UM	CREATE.inv
sc-id	—	by UM	CREATE.inv
production-state	Production (5.10)	= *waiting*	—
data-units-consumed-for-processing	Production (5.10)	= 0	as needed
processed-successfully	Production (5.10)	= 0	as needed
data-units-produced	Production (5.10)	= 0	as needed
number-invocations	Production (5.10)	= 0	as needed
unavailable-time	Production (5.10)	= 0	as needed
operational-time	Production (5.10)	= 0	as needed
production-start-time	Production (5.10)	by CM	when production starts
production-stop-time	Production (5.10)	by CM	when production stops
done-time	Production (5.10)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

12.7.6 ACTIONS—F-vc-multiplex-prod MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the Production managed object class.

12.7.7 STATE-RELATED BEHAVIOR—F-vc-multiplex-prod MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the Production managed object class.

12.7.8 NOTIFICATIONS—F-vc-multiplex-prod MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the Production managed object class.

12.7.9 VALIDATION ASPECTS—F-vc-multiplex-prod MO CLASS

No specific validation aspects are defined for this managed object class. However, validation aspects are inherited from the Production managed object class.

12.8 F-vc-seg-prod MANAGED OBJECT CLASS

12.8.1 PURPOSE

This managed object embodies the management interface to the functionality that constructs TC Frame Data Units from packets or TC segments. This production is executed as part of the Forward TC-VC-Data Insertion functional group.

12.8.2 INHERITANCE—F-vc-seg-prod MO CLASS

This managed object class is derived from and inherits the properties of the Production managed object class.

12.8.3 OBJECTS CONTAINED—F-vc-seg-prod MO CLASS

An instance of this concrete managed object class shall contain instances of the following classes:

a) Zero, one, or more f-sp-ts-p managed objects. This class is specified in 12.2.

b) Zero, one, or more f-sp-st managed objects. This class is specified in 12.1.

12.8.4 STATES—F-vc-seg-prod MO CLASS

No specific states are defined for this managed object class. However, states are inherited from the Production managed object class.

12.8.5 ATTRIBUTES—F-vc-seg-prod MO CLASS

This managed object class has the following specific attributes:

a) **f-vc-seg-prod-mo-id.** The RDN of the managed object used to construct an unambiguous DN for the instance of this managed object class (refer to annex C).

   1) Data type/values: Character string.

   2) Modifications: SET shall be disabled in both the definition and utilization phases.

b) **map-multiplexing-scheme.** The MAP multiplexing scheme for the virtual channel.

   1) Data type/values:

      i) 'fifo'. The requests are served first-in-first-out.

      ii) 'absolute-priority'. The highest priority request is always served first.

      iii) 'polling-vector'. Requests are served according to a predefined polling vector.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

c) **max-tc-user-data-unit-length.** The maximum length (in number of octets) of the TC user data units that each virtual channel can accept.

   1) Data type/value: Unsigned integer.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

d) **packet-version-supported.** The CCSDS packet version that is allowed to be sent on the virtual channel.

   1) Data type/value: Unsigned integer.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

e) **segmentation-flag.** The segment header is/is not to be present in the TC Frame Data Unit produced by the virtual channel.

   1) Data type/values:

      i) 'segment-header-present'. Segmentation is required and MAPs are used.

      ii) 'no-segmentation-applied'. Segmentation is not required.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

f) **map-multiplexing-control.** The priority list or polling vector in effect.

   1) Data type/values: See item 3 below.

   2) Modifications: SET shall be enabled when the managed object is in the state *waiting*.

   3) Dependencies: The attribute value depends on the value of *map-multiplexing-scheme* as described above.

      i) If the value of *map-multiplexing-scheme* is 'fifo', the value of *map-multiplexing-control* shall be 'null'.

      ii) If the value of *map-multiplexing-scheme* is 'absolute priority', the value of *map-multiplexing-control* shall be a sequence of up to 64 MAP-IDs, there MAP-ID is an integer of value 0-63. The first value in the sequence shall be the MAP-ID of the highest-priority MAP, the second value shall be the MAP-ID of the second-highest priority MAP, etc.

      iii) If the value of *map-multiplexing-scheme* is 'polling vector', the value of *map-multiplexing-control* shall be a sequence of up to 192 MAP-IDs, where MAP-ID is an integer of value 0-63. The first value in the sequence shall be the

MAP-ID of the MAP to be polled first, the second value shall be the MAP-ID of the MAP to be polled second, etc.

NOTE – A MAP-ID may appear more than once in the polling vector (e.g., in the case where a high-data-rate MAP must be polled more often than other lower-rate MAPs).

g) **initial-map-multiplexing-control.** The initial value of *map-multiplexing-control.*

1) Data type/values: See item f) 3) above.

2) Modifications: SET shall be disabled in both the definition and utilization phases.

3) Dependencies: Possible attribute values depend on the value of *map-multiplexing-scheme* described above.

h) **Attribute Value Summary**. Refer to table 12-6.

Table 12-6: Attribute Value Summary—F-vc-seg-prod Managed Object

Attribute	Inherited From	Initial Value	Specified in
f-vc-seg-prod-mo-id	—	by CM	CREATE.inv
map-multiplexing-scheme	—	by CM	CREATE.inv
max-tc-user-data-unit-length	—	by CM	CREATE.inv
packet-version-supported	—	by CM	CREATE.inv
segmentation-flag	—	by CM	CREATE.inv
map-multiplexing-control	—	by CM	CREATE.inv
initial-map-multiplexing-control	—	by CM	CREATE.inv
production-state	Production (5.10)	= *waiting*	—
data-units-consumed-for-processing	Production (5.10)	= 0	as needed
processed-successfully	Production (5.10)	= 0	as needed
data-units-produced	Production (5.10)	= 0	as needed
number-invocations	Production (5.10)	= 0	as needed
unavailable-time	Production (5.10)	= 0	as needed
operational-time	Production (5.10)	= 0	as needed
production-start-time	Production (5.10)	by CM	when production starts
production-stop-time	Production (5.10)	by CM	when production stops
done-time	Production (5.10)	by CM	when MO enters *done*
read-me	SLEManagementEntity (5.18)	by UM	CREATE.inv

12.8.6 ACTIONS—F-vc-seg-prod MO CLASS

No specific actions are defined for this managed object class. However, actions are inherited from the Production managed object class.

12.8.7 STATE-RELATED BEHAVIOR—F-vc-seg-prod MO CLASS

No specific state-related behavior is defined for this managed object class. However, state-related behavior is inherited from the Production managed object class.

12.8.8 NOTIFICATIONS—F-vc-seg-prod MO CLASS

No specific notifications are defined for this managed object class. However, notifications are inherited from the Production managed object class.

12.8.9 VALIDATION ASPECTS—F-vc-seg-prod MO CLASS

No specific validation aspects are defined for this managed object class. However, notifications are inherited from the Production managed object class.

13 FORWARD AOS SPACE LINK PROCESSING MANAGED OBJECT DEFINITIONS

NOTE – Although the *Cross Support Reference Model* (reference [1]) specifies a set of forward AOS services and functional groups, the current suite of SLE services do not contain any forward AOS services. These managed object class definitions are placeholders for those that will be needed when forward AOS services are specified.

13.1 F-c/vcdu-prod MANAGED OBJECT CLASS

13.1.1 PURPOSE

This managed object embodies the management interface to the functionality that produces C/VCDUs from protoVCDUs. This process includes adding the appropriate error correction fields and insert zones (if present). This production is executed as part of the Forward AOS Space Link Processing functional group.

13.1.2 INHERITANCE—F-c/vcdu-prod MO CLASS

This managed object class is derived from and inherits the properties of the Production managed object class.

13.2 F-c/vcdu-st MANAGED OBJECT CLASS

13.2.1 PURPOSE

This managed object represents the resources needed to store a forward C/VCDU channel. Refer to 5.3 for general storage considerations.

13.2.2 INHERITANCE—F-c/vcdu-st MO CLASS

This managed object class is derived from and inherits the properties of the ChannelStored managed object class.

13.3 F-c/vcdu-ts-p MANAGED OBJECT CLASS

13.3.1 PURPOSE

This managed object represents the resources needed to provide a forward C/VCDU service. Refer to 5.20 for a general discussion of transfer service provision.

13.3.2 INHERITANCE—F-c/vcdu-ts-p MO CLASS

This managed object class is derived from and inherits the properties of the F-transferService managed object class.

13.4 F-insert-st MANAGED OBJECT CLASS

13.4.1 PURPOSE

This managed object represents the resources needed to store a forward bitstream channel. Refer to 5.3 for general storage considerations.

13.4.2 INHERITANCE—F-insert-st MO CLASS

This managed object class is derived from and inherits the properties of the ChannelStored managed object class.

13.5 F-insert-ts-p MANAGED OBJECT CLASS

13.5.1 PURPOSE

This managed object represents the resources needed to provide a forward insert transfer service. Refer to 5.20 for a general discussion of transfer service provision.

13.5.2 INHERITANCE—F-insert-ts-p MO CLASS

This managed object class is derived from and inherits the properties of the F-transferService managed object class.

13.6 F-pca-pdu-prod MANAGED OBJECT CLASS

13.6.1 PURPOSE—F-pca-pdu-prod MO CLASS

This managed object embodies the management interface to the functionality that combines one or more C/VCDU data channels into a single AOS physical channel access protocol data unit (PCA-PDU) (refer to *AOS,* reference [5]). This production is executed as part of the Forward AOS Space Link Processing functional group.

13.6.2 INHERITANCE—F-pca-pdu-prod MO CLASS

This managed object class is derived from and inherits the properties of the Production managed object class.

13.7 F-proto-vcdu-ts-p MANAGED OBJECT CLASS

13.7.1 PURPOSE

This managed object represents the resources needed to provide a forward protoVCDU transfer service. Refer to 5.20 for a general discussion of transfer service provision.

13.7.2 INHERITANCE—F-proto-vcdu-ts-p MO CLASS

This managed object class is derived from and inherits the properties of the F-transferService managed object class.

14 FORWARD AOS VC DATA INSERTION MANAGED OBJECT DEFINITIONS

NOTE – Although the *Cross Support Reference Model* (reference [1]) specifies a set of forward AOS services and functional groups, the current suite of SLE services do not contain any forward AOS services. These managed object class definitions are placeholders for those that will be needed when forward AOS services are specified.

14.1 F-bitstream-st MANAGED OBJECT CLASS

14.1.1 PURPOSE

This managed object represents the resources needed to store a forward bitstream channel. Refer to 5.3 for general storage considerations.

14.1.2 INHERITANCE—F-bitstream-st MO CLASS

This managed object class is derived from and inherits the properties of the ChannelStored managed object class.

14.2 F-bitstream-ts-p MANAGED OBJECT CLASS

14.2.1 PURPOSE

This managed object represents the resources needed to provide a forward bitstream transfer service. Refer to 5.20 for a general discussion of transfer service provision.

14.2.2 INHERITANCE—F-bitstream-ts-p MO CLASS

This managed object class is derived from and inherits the properties of the F-transferService managed object class.

14.3 F-proto-vcdu-b_pdu-prod MANAGED OBJECT CLASS

14.3.1 PURPOSE

This managed object embodies the management interface to the functionality that consumes one bitstream channel, forms B_PDUs (refer to *AOS,* reference [5]), and produces protoVCDUs. This production is executed as part of the Forward AOS VC Data Insertion functional group.

14.3.2 INHERITANCE—F-proto-vcdu-b_pdu-prod MO CLASS

This managed object class is derived from and inherits the properties of the Production managed object class.

14.4 F-proto-vcdu-m_pdu-prod MANAGED OBJECT CLASS

14.4.1 PURPOSE

This managed object embodies the management interface to the functionality that consumes one or more channels of space packets, forms M_PDUs (refer to *AOS,* reference [5]), and produces protoVCDUs. This production is executed as part of the Forward AOS VC Data Insertion functional group.

14.4.2 INHERITANCE—F-proto-vcdu-m_pdu-prod MO CLASS

This managed object class is derived from and inherits the properties of the Production managed object class.

14.5 F-proto-vcdu-st MANAGED OBJECT CLASS

14.5.1 PURPOSE

This managed object represents the resources needed to store a protoVCDU channel. Refer to 5.3 for general storage considerations.

14.5.2 INHERITANCE—F-proto-vcdu-st MO CLASS

This managed object class is derived from and inherits the properties of the ChannelStored managed object class.

14.6 F-proto-vcdu-vca-prod MANAGED OBJECT CLASS

14.6.1 PURPOSE

This managed object embodies the management interface to the functionality that consumes VCA-SDUs (refer to *AOS,* reference [5]) and produces protoVCDUs. This production is executed as part of the Forward AOS VC Data Insertion functional group.

14.6.2 INHERITANCE—F-proto-vcdu-vca-prod MO CLASS

This managed object class is derived from and inherits the properties of the Production managed object class.

14.7 F-proto-vcdu-ts-u MANAGED OBJECT CLASS

14.7.1 PURPOSE

This managed object represents the resources needed to use a forward protoVCDU transfer service. Refer to 5.20 for a general discussion of transfer service use.

14.7.2 INHERITANCE—F-proto-vcdu-ts-u MO CLASS

This managed object class is derived from and inherits the properties of the F-transferService managed object class.

14.8 F-vca-st MANAGED OBJECT CLASS

14.8.1 PURPOSE

This managed object represents the resources needed to store a forward VCA channel. Refer to 5.3 for general storage considerations.

14.8.2 INHERITANCE—F-vca-st MO CLASS

This managed object class is derived from and inherits the properties of the ChannelStored managed object class.

14.9 F-vca-ts-p MANAGED OBJECT CLASS

14.9.1 PURPOSE

This managed object represents the resources needed to provide a VC Access transfer service. Refer to 5.20 for a general discussion of transfer service provision.

14.9.2 INHERITANCE—F-vca-ts-p MO CLASS

This managed object class is derived from and inherits the properties of the F-transferService managed object class.

ANNEX A

SLE SERVICE MANAGEMENT ACRONYMS

(This annex is not part of the draft Recommendation.)

AOS	Advanced Orbiting System
APID	Application Process Identifier
ASCII	American Standard Code for Information Interchange
ASDC	Abstract Service Definition Conventions
ASN.1	Abstract Syntax Notation
CCSDS	Consultative Committee for Space Data Systems
CLCW	Command Link Control Word
CLTU	Command Link Transmission Unit
CM	SLE Complex Management
CMIS/CMIP	Common Management Information Service/Protocol
CMM	Carrier Modulation Mode
COP	Command Operation Procedure
CORBA	Common Object Request Broker Architecture
COTS	Commercial Off-the-shelf (products)
C/VCDU	Coded Virtual Channel Data Unit
DCE	Distributed Computing Environment
DN	Distinguished Name
ERT	Earth Receive Time
FAF	Forward All Frames
FECF	Frame Error Control Field
FIFO	First-In, First-Out
FSH	Frame Secondary Header
FSP	Forward Space Packet
FTCF	Forward TC Frames

GDMO	Guidelines for Definition of Managed Objects
GVCID	Global Virtual Channel Identifier
ISO	International Organization for Standardization
LIFO	Last-In, First-Out
MAP	Multiplexer Access Point
MC	Master Channel
MDOS	Mission Data Operation System
MO	Managed Object
OCF	Operational Control Field
OID	Object Identifier
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
PLOP	Physical Layer Operations Procedure
PT	Packet Telemetry
PTTF	Packet Telemetry Transfer Frame
RAF	Return All Frames
RCF	Return Channel Frames
RDN	Relative Distinguished Name
RF	Radio Frequency
RSP	Return Space Packet
SC	Spacecraft
SCID	Spacecraft Identification
SIB	Security Information Base
SLE	Space Link Extension
SLE-SDU	Space Link Extension Service Data Unit
SNMP	Simple Network Management Protocol
TC	Telecommand
TCM	Terminology, Conventions, and Methodology

TIB	Telecommunication Information Base
UM	SLE Utilization Management
UML	Unified Modeling Language
URN	Uniform Resource Name
VC	Virtual Channel
VCA	Virtual Channel Access
VCDU	Virtual Channel Data Unit
VCID	Virtual Channel Identifier

ANNEX B

GLOSSARY

(This annex is not part of the draft Recommendation.)

This annex defines terminology used in this draft Recommendation. Many of the terms used in this document have been taken from other sources, which are listed in 1.8.

Complex Management .... The management entity within the SLE Complex that supports the spacecraft.

Debrief... Period(s) during the Utilization Phase of the service package lifetime during which the SLE Complex(es) and SLE Utilization Management exchange information about the SLE services provided.

Definition Phase .... Period in the service package lifetime during which the servicePackage managed object and its contained managed objects are created.

Functional Group.. An abstraction of Complex resources that process SLE data channels as needed to deliver SLE transfer services.

Managed Object (MO) .... An abstraction of a resource that represents that resource's properties as seen by (and for the purposes of) SLE service management. A managed object is defined in terms of its attributes, the operations that may be performed on the object, the notifications that it may issue, and its relationships to other managed objects.

Managed Object Attribute............................ An attribute represents a property of the resource represented by a managed object. Attributes have values that are used to configure, monitor, or control the resource.

Managed Object Instance .... One specific occurrence of a given managed object class. One class may have many instances, but each instance is of only one class.

Managed Object Class.................................. A template for managed object instances that have the same attributes, operations and notifications. One class may have many instances.

Mission Data Operations System (MDOS)... The entity requiring SLE transfer services. The MDOS encompasses SLE Utilization Management.

Mission User Entity...................................... Entities within the MDOS that use specific SLE transfer services. Mission User Entities may be at different physical facilities.

Partition.. A portion of the Complex's storage resource that accommodates a single type of SLE data channel.

Service Agreement....................................... An agreement between two or more entities to provide SLE services. The Service Agreement includes identification of the space mission and SLE Complex, definition of services to be provided and terms of provision, and the term of the agreement itself.

Service Agreement Period............................ Time during which an SLE Complex provides the capability to create SLE service packages in accordance with the SLE Service Agreement.

Service Instance.. The capability of transferring one or more SLE data channels of a given type. Also referred to as transfer service instances.

Service Instance Provision Period................ Period of time during which the Complex is capable of transferring one or more SLE data channels of a given type.

Service Package.. A mission's set of requirements for SLE transfer services. These services would be provided by a single Complex over a specific period.

Service Package Lifetime............................. Time during which the Complex processes an SLE service package and the Service Instances exist.

SLE Entity .... An SLE application process that either uses or provides a single instance of SLE transfer service.

Transfer Service .... Transfer services transfer SLE data channels between a Space Element and an MDOS through the SLE system and SLE Management Service.

Utilization Management................................ Entity within an MDOS that interfaces with the Complex for use of SLE services.

Utilization Phase .... Portion of the service package lifetime during which the Complex provides the negotiated SLE transfer services.

ANNEX C

SLE NAMING CONVENTIONS

(This annex is part of the draft Recommendation.)

C1 SLE DATA CHANNELS

The concept of space data channels, introduced in the *Cross Support Reference Model* (reference [1]), is central to SLE service management. A space data channel is a stream of space data units of the same type, associated with each other by a single, unique identifier. There are two sets of space data channels:

a) **Space Link Data Channel.** A space link data channel carries space link data units between the SLE system and the space element.

b) **SLE Data Channel.** The SLE Data Channel is specific to the SLE system and carries SLE Service Data Units (SLE-SDUs) between the components of the SLE system, and between the SLE system and the MDOS.

This annex expands on this concept to provide the means to specify the space data channel from which a specific SLE Service Instance is derived.

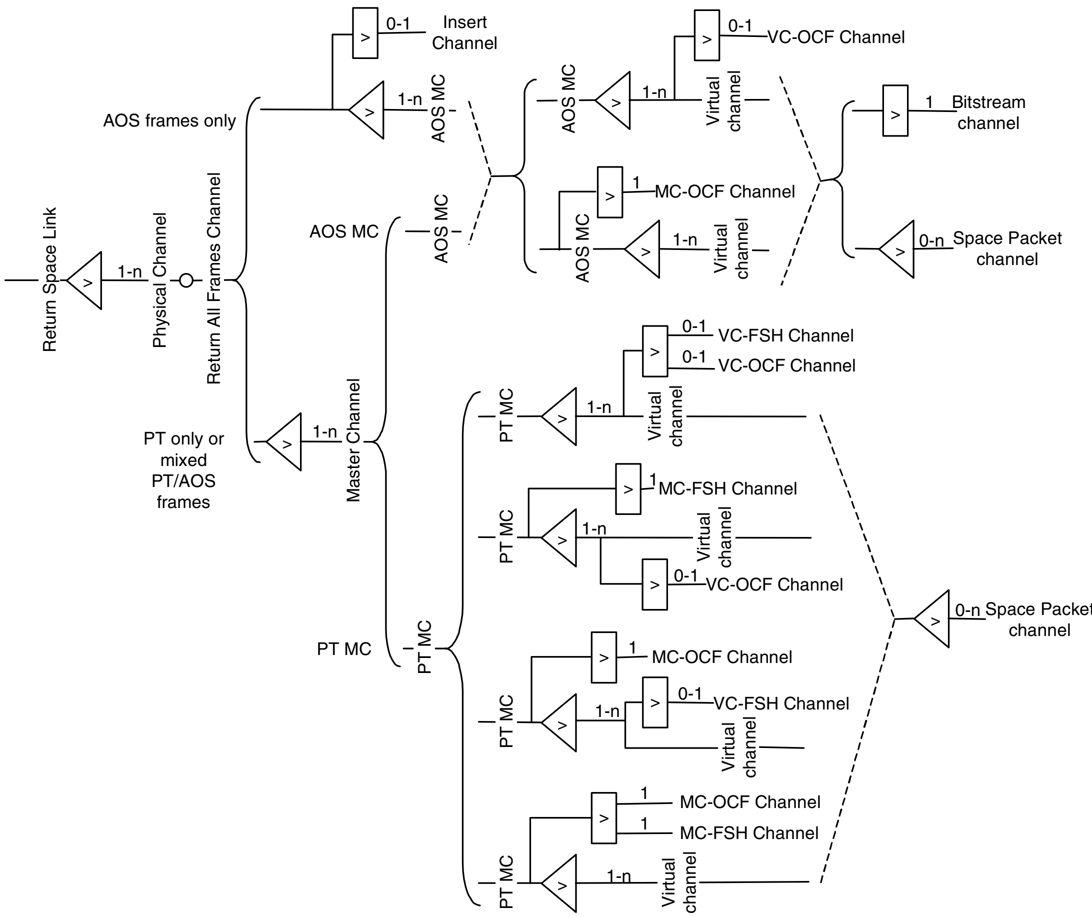
C2 SLE DATA CHANNEL TREES

The *Cross Support Reference Model* (reference [1]) also introduced the concept of space data channel trees.

a) **Space Data Channel Tree.** Space data channel trees are hierarchical structures that relate the various space data channels to one another.

b) **SLE Data Channel Tree.** SLE data channel trees are specific to the SLE system and represent the hierarchical structure of CCSDS Space Link protocol data units and the SLE services derived from them.

Figures C-1 and C-2 illustrate the return and forward SLE data channel trees. Together, these channel trees represent all types of SLE data channels that may be provided by an SLE Complex to a mission.



KEY

Decommutator (in return direction). Each output line (line to the right of the rectangle) indicates one channel of a given type. The number of lines corresponds to the number of different channel types. The number above the output line indicates the number of instances of that type that are possible. Example values are:

0-n	zero or more instances
1-n	one or more instances
1	exactly one instance

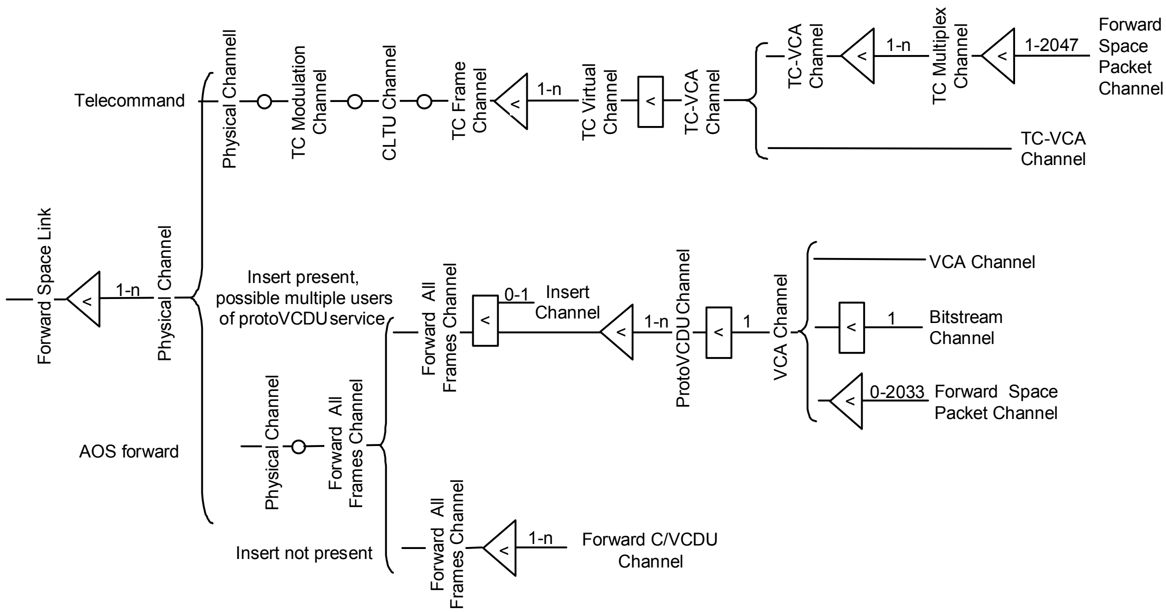
Demultiplexer (in return direction). The output line (line to the right of the triangle) indicates one channel of the same type. The number above the output line indicates the number of instances of that type that are possible (same values as above).

OR. A channel to the left of the OR symbol may consist of any of the subtree types to the right of the symbol.

Synchronous physical channel processing

Tap-off

Figure C-1: Return SLE Data Channel Tree



KEY

a, b — Commutator (in forward direction). Each input line (line to the right of the rectangle) indicates one channel of a given type. The number of lines corresponds to the number of different channel types. The number above the input line indicates the number of instances of that type that are possible. Example values are:

0-n	zero or more instances
1-n	one or more instances
1	exactly one instance

a — Multiplexer (in forward direction). The input line (line to the right of the triangle) indicates one channel of the same type. The number above the input line indicates the number of instances of that type that are possible (same values as above).

{ OR. A channel to the left of the OR symbol may consist of any of the subtree types to the right of the symbol.

O Physical channel processing. In the case of Telecommand, this is the generation of the Physical Layer stream (Acquisition Sequence, Idle Sequence, and injected CLTUs). In the case of AOS, this is the attachment of sync markers to, and the synchronous injection of, C/ VCDUs

Figure C-2: Forward SLE Data Channel Tree

C3 SLE SPACECRAFT CHANNEL PROFILES

a) An SLE Data Channel Profile is a subset of the SLE data channel trees (return and forward) that applies to a specific spacecraft.

b) The CCSDS Space Link Recommendations identify many space data channels that a spacecraft can use. The SLE Data Channel Tree presents the full set of these space data channels. The SLE Data Channel Profile presents only the space link extension data channels that a particular mission's spacecraft is actually designed to use. Figure

C-3 shows an example of an SLE Data Channel Profile. (Compare to full SLE data channel tree in figure C-1.)

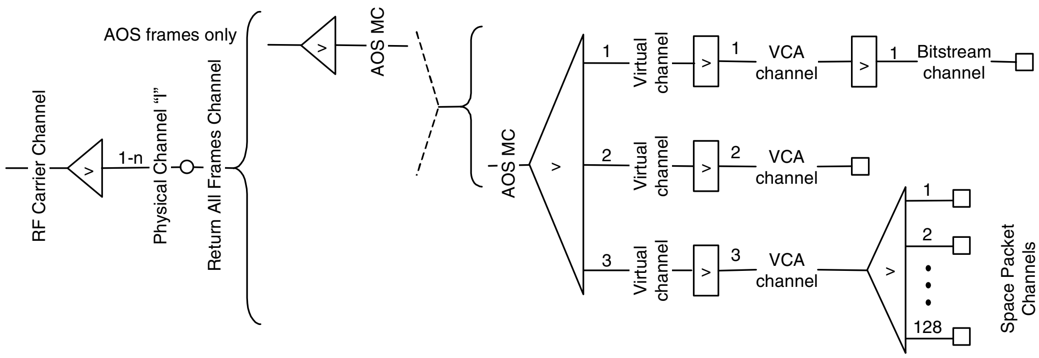


Figure C-3: Example: Mission XYZ Return Data Channel Profile

C4 SLE DATA CHANNEL NAMES

SLE data channels play a fundamental role in specifying SLE service agreements and SLE service packages. In order to identify which SLE data channels are used to provide specific SLE service instances, these SLE data channels must be unambiguously named.

C4.1 SLE Data Channel Name Hierarchy

Figure C-4 presents the name hierarchy for SLE Data Channels. This diagram resembles the containment diagram presented for SLE data channels in figures C-1 and C-2, but instead of containment, it shows how names for instances of SLE data channels are created. In this case, the natural containment of the SLE data channels is used to name those data channels. However, the containment relationships shown here differ somewhat from those of the SLE data channel trees, in that redundancies are eliminated for naming purposes. For example, the return SLE data channel tree shows the return OCF (ret_OCF channel as being 'contained' within the return channel frame (ret_CF) channel. However, since both the ret_OCF and the ret_CF channels are identified by the same identifier, the GVCID appears only once in the name hierarchy.

Not all SLE data channels are named. SLE data channel names are only necessary when the channel is exposed at an SLE transfer service port, because the data channel name is required to form the name of a data channel that is exposed at an external interface, or to link a particular portion of the data channel profile to a symbol stream. Two of the SLE data channels that appear in Figures C-1 and/or C-2 are not named: the physical channel and the TC Virtual channel. Both of these channel types are strictly internal to SLE Complexes and are never exposed at external interfaces.

To distinguish this streamlined naming hierarchy from the pure containment of the channel tree, the 'is named by' relationship is used to relate these data channels for naming purposes. In the diagram, the 'is named by' relationship is designated by a line terminating at a small black box, such that the name-attribute that is marked with the small black box is named by the entity at the other end of the line. For example, the ret_CF channel 'is named by' the return all frames (rAF) channel.

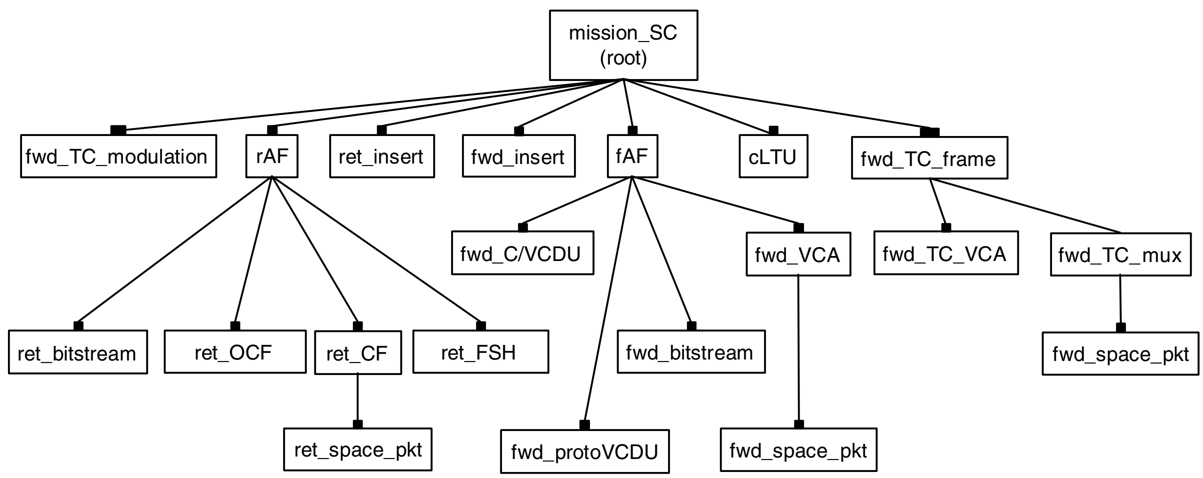


Figure C-4: SLE Data Channel Name Hierarchy

C4.2 SLE Data Channel Naming Rules

The naming scheme for SLE data channels is based on the naming scheme used in OSI management (*OSI—Basic Reference Model,* reference E5). Since an SLE data channel is naturally contained within other SLE data channels, the name of a specific SLE data channel is formed by the iterative concatenation of the names of all of the channels that contain it. Therefore, the name of an instance of a data channel is formed as follows:

a) Relative distinguished name (RDN): For each type of data channel (e.g., return all frames, return frame channel) there is a corresponding naming attribute (e.g., rAF, ret_CF). The RDN of an instance of a data channel is formed by combining the data channel naming attribute with a value that identifies the instance. The RDN of the SLE data channel instance must be unique within the scope of the containing SLE data channel. For example, only one ret_space_pkt channel #2 can be contained within a particular return frame channel (ret_CF), but a different return frame channel for the same spacecraft also may have a ret_space_pkt channel #2.

b) Distinguished name (DN): The DN for a particular SLE data channel is formed by concatenating the RDN of the SLE data channel to the DN of the SLE data channel that contains it.

The *root* of the SLE data channel name is the DN of the mission spacecraft (mission_SC) for which the channel tree exists. The syntax of the mission_SC RDN is an IA5String, and the value is the same as the spacecraft name used in the construction of the value of the *service-agreement-mo-id* value (refer to C5).

A real mission spacecraft can have any number of forward and return symbol streams, depending on the type, number, and technology of the RF and modulation equipment used by the spacecraft. Each symbol stream carries one physical channel. For each forward physical channel that carries Telecommand-formatted data, there is one associated TC Modulation channel, one associated CLTU data channel, and one associated TC frame data channel. For each forward physical channel that carries AOS-formatted data, there is one associated FAF data channel and optionally one forward Insert data channel. For each return physical channel, there is one associated RAF data channel and optionally one return Insert data channel. Thus, the mission_SC can name (i.e., contain) any number of physical channel-associated SLE data channels (rAF, ret_insert, FAF, fwd_insert, fwd_TC_modulation, cLTU, fwd_TC_Frame).

The values used for the RDNs of the physical channel-associated data channels are arbitrary, subject to the following constraints:

a) For each return physical channel, there is one and only one rAF channel. The syntax of the RDN value of the RAF channel is IA5String.

b) For each rAF channel, there is zero or one ret_insert channel. If a ret_insert channel is present, the RDN value of the ret_insert channel is the same as the RDN value of the associated rAF channel.

c) For each forward physical channel, there is either one fAF channel or one TC Modulation channel. The syntax of the RDN value of the fAF or TC Modulation channel is IA5String.

d) For each fAF channel, there is zero or one fwd_insert channel. If a fwd_insert channel is present, the RDN value of the fwd_insert channel is the same as the RDN value of the associated fAF channel.

e) For each TC Modulation channel, there is one CLTU data channel. The RDN value of the CLTU channel is the same as the RDN value of the associated TC Modulation channel.

f) For each CLTU channel, there is one fwd_TC_Frame data channel. The RDN value of the fwd_TC_Frame channel is the same as the RDN value of the associated CLTU channel.

NOTE – Although the values of the physical channel-associated data channel RDNs are arbitrary, missions will most likely give them names that have some mission significance, e.g., ‘high-rate_science’ or ‘housekeeping.’

For the other SLE data channels, the values of their RDNs are equal to the identifiers found in their respective CCSDS protocol data units.

Table C-1 defines the syntax of the values for the SLE data channel naming attributes shown in figure C-4. In some cases, the same syntax provides values for two or more naming attributes. For instance, an APID (as defined in the *Cross Support Reference Model,* reference [1]) provides the value for either a ret_space_packet channel instance, or a fwd_space_packet channel instance.

Table C-1: SLE Data Channel Naming Attributes & Syntax

SLE Data Channel Naming Attribute	Attribute Value Syntax
mission-SC	IA5String
fwd-tc-mod_ch	IA5String
rAF	IA5String
ret_insert	IA5String
ret_bitstream	GVCID = frame version number + SCID + VCID*
ret_CF	GVCID = frame version number + SCID + VCID*
ret_OCF	GVCID = frame version number + SCID + VCID*
ret_FSH	GVCID = frame version number + SCID + VCID*
ret_space_pkt	APID
fAF	IA5String
fwd_insert	IA5String
fwd_C/VCDU	GVCID = frame version number + SCID + VCID
fwd_proto-VCDU	GVCID = frame version number + SCID + VCID
fwd_VCA	GVCID = frame version number + SCID + VCID
fwd_bitstream	GVCID = frame version number + SCID + VCID
cLTU	IA5String
fwd_TC_Frame	IA5String
fwd_TC_VCA	GVCID = frame version number + SCID + VCID**
fwd_TC_mux	Global VCID = frame version number + SCID + VCID
fwd_space_pkt	APID

* A return frame channel can be either an individual virtual channel or a predefined aggregation of virtual channels known as a master channel. In the case of a single virtual channel, the value of the VCID portion of the GVCID is a single value in the range (0..63). In the case of a master channel, the value of the VCID portion of the GVCID is a single value 'all'.

** TC VCA channels contained in a single fwd_TC_Frame channel must have the same frame version number and SCID.

Each GVCID has the form of a triplet (version number, SCID, VCID), where the valid combinations of values are listed in table C-2.

Table C-2: GVCID Triplets

Version Number	SCID Range	VCID Range
1	0 - 1023	0 - 7
2	0 - 255	0 - 63

C4.3 Example of SLE Data Channel Naming

As an example of SLE data channel naming, consider the mission ‘XYZ’, which has a radio system capable of generating signals that can carry a number of return physical channels. One of these physical channels carries a return all frames channel bearing high-rate science data, and is designated ‘HiRateScience’. The high-rate science data RAF channel carries several master channels, one of which is the Packet Telemetry-formatted (frame version number 1) MC#4. MC#4 carries several VCs, one of which is VC#25. VC#25 carries multiple space packet channels, one of which is identified by APID#47.

The SLE data channel names are built as follows:

Table C-3: SLE Data Channel Names

SLE Data Channel	Relative Distinguished Name	Distinguished Name
Mission Spacecraft XYZ	mission_SC	mission_SC = ‘XYZ’
Return All Frames channel for High Rate Science physical channel	rAF = ‘HiRateScience’	mission_SC = ‘XYZ’, rAF = ‘HiRateScience’
Master channel # 4	ret_CF = (1,4,all)	mission_SC = ‘XYZ’, rAF = ‘HiRateScience’ ret_CF = (1,4,all)
Virtual channel # 25	ret_CF = (1,4,25)	mission_SC = ‘XYZ’, rAF = ‘HiRateScience’ ret_CF = (1,4,25)
Return Space Packet Channel # 47	ret_space_pkt = 47	mission_SC = ‘XYZ’, rAF = ‘HiRateScience’ ret_CF = (1,4,25) ret_space_pkt = 47

C5 SLE MANAGED OBJECT NAMES

Since there may be many instances of an SLE managed object class, each instance of an SLE managed object must be identified by a unique name, so that management operations can be carried out without ambiguity. The method for creating names for the managed objects is based on the managed object containment relationship.

To enable both SLE Utilization Management and SLE Complex Management to identify the managed object that is to be affected by a management operation, unambiguous naming of SLE managed object instances is necessary. The naming of SLE managed objects is based on the naming scheme used in the OSI management.

The specification of SLE managed objects names has two aspects: the construction of the names (that is, how the names are logically formed), and the representation of the names (how the names are 'encoded' for exchange).

C5.1 SLE Managed Object Name Construction

SLE managed object names are constructed using the containment diagram for SLE managed objects (refer to annex D).

Each concrete SLE managed object class (e.g., ServicePackage) has a corresponding naming attribute. For each SLE managed object class this naming attribute is unique, but by the convention adopted for this draft Recommendation is of the form 'xxx-mo-id', where 'xxx' is an abbreviation of the managed object class name. The exact names of the naming attributes are specified in their respective managed object definitions.

The RDN of an instance of an SLE managed object is formed by pairing the naming attribute of the managed object class with a value that distinguishes that instance from other managed object instances.

The DN for of an instance of an SLE managed object is formed by appending the managed object's RDN to the end of the DN of its containing managed object. Since the method of constructing the name applies to all objects, the containing managed object is named in the same way.

The name of the serviceAgreement managed object instance for a particular cross support agreement is used to name its contained objects, and serves as the root of the names of all SLE managed objects associated with that Service Agreement. The name of an SLE Service Agreement must be unique within both the supporting Agency (owner of the SLE Complex) and the supported agency (owner of the MDOS). Assuming that there are CCSDS-unique names for a spacecraft as well as unique acronyms for the Agencies involved in cross support, the value of the RDN of a serviceAgreement managed object instance is formed from the acronyms for the names of the supporting Agency and supported Agency, the spacecraft name, and a number to distinguish between revisions of or alternate SLE service agreement instances:

service-agreement-mo-id value =<agencies>**:**<complex>**-**<spacecraft>**-**(<number>).

Where:

a) <agencies> is composed of the acronyms of the supporting Agency and the supported Agency, in that order, separated by a hyphen ('-') form the first part of the name.

b) <complex> is the name of the SLE Complex that will provide SLE services.

c) <spacecraft> is the name of the supported spacecraft.

d) <number> is a number (e.g., serial number) assigned by the supporting Agency to distinguish among revisions or alternate SLE Service Agreement instances.

These identifiers are assigned in accordance with *CCSDS Control Authority Procedures* (reference [E11]).

NOTE – For example, a NASDA mission with a spacecraft named 'N22' supported by a CNES SLE Complex named 'CMX88', might have a serviceAgreement managed object with the name:

service-agreement-mo-id=CNES-NASDA:CMX88-N22-(01).

Joint missions could use a compound Agency name, e.g., NASDA/DLR.

C5.2 SLE Managed Object Name Representation

Two methods are used for representing SLE managed object names: Shorthand and Object Identifier. The following subsections describe each of these representations in turn.

C5.2.1 Shorthand Representation

Shorthand representation is intended to be 'human readable' and 'human typable.' The emphasis is on a representation that minimizes keystrokes while maintaining understandability.

In the shorthand form, the naming attribute identifier is represented by character strings, which cannot contain '.', ',' or '='. The naming attribute identifiers for all of the SLE managed objects are listed (with all of the SLE management attributes) in *Service Management-Managed Object Formal Specification* (reference [10]). The naming attributes all have labels that end in 'mo-id'.

The naming attribute value is represented by a typable character string. The character string cannot contain white space or '='.

The RDN of an object is represented by joining the identifier and the value with '='. For the example given in D5.1, where *sagr* is the shorthand naming attribute identifier and 'CNES-NASDA:CMX88-N22-(01)' the value of a particular serviceAgreement instance, the RDN is represented as:

sagr = `CNES-NASDA:CMX88-N22-(01)`

The DN is represented by the hierarchically-ordered series of RDNs, separated by commas (‘,’). For example, consider an instance of a return all frames ‘service used’ managed object (shorthand naming attribute identifier *raf-u*) under the 98th service package (shorthand naming attribute identifier *spack*) of the service agreement named above, as illustrated in figure C-5. Assuming that the value of the service package is ‘98’, and the value of the r-af-ts-u instance is designated as '01' (which is unique within service package #98), the shorthand representation of the DN for the service instance is:

sagr = `CNES-NASDA:CMX88-N22-(01)`, spack = `98`, raf-u = `01`

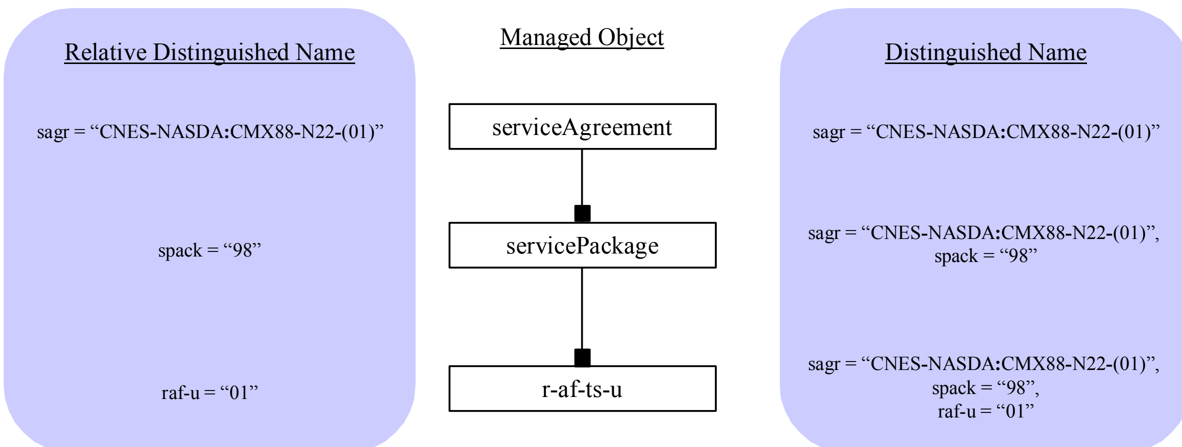


Figure C-5: Example: SLE Shorthand Managed Object Name Construction

C5.2.2 Object Identifier (OID) Representation

Object Identifier representation is suited to machine processing and encoding for transfer. The ISO-defined Object Identifier representation uses a sequence of integers to uniquely represent each naming attribute identifier. Each such sequence is registered with a registration authority that is properly authorized in accordance with ISO procedures to assign object identifiers to objects requiring identification, such as naming attribute identifiers. When an object is properly registered by such an authority, the resulting OID is uniquely assigned to that object for all ISO-conformant installations. annex D presents a description of how these sequences of integers are structured in accordance with the hierarchically-arranged ISO registration tree. In the Object Identifier representation, the naming attribute identifier is represented by the sequence of integers that constitute the OID, separated by spaces and surrounded by curly braces ('{', '}'). The naming attribute value is represented by a typable character string without white space. The RDN of an object is represented by enclosing the identifier and the value with square brackets ('[', ']'). The DN is represented by the hierarchically-ordered series of RDNs.

In the following examples, the object identifiers are as defined in *Service Management-Managed Object Formal Specification* (reference [10]). *Service-agreement-mo-id* (the naming attribute for the service agreement) is registered as 1 3 112 4 3 2 156.

For the example of the service agreement RDN given in D5.2.1, the RDN of the service agreement is represented as:

[{1 3 112 4 3 2 156} `CNES-NASDA:CMX88-N22-(01)`]

For the example in C5.2.1, the corresponding Object Identifier representation for the name of the RAF transfer 'service used' instance would be:

[{1 3 112 4 3 2 156} `CNES-NASDA:CMX88-N22-(01)`]
[{1 3 112 4 3 2 176} `98`]
[{1 3 112 4 3 2 115} `01`]

where the attribute *service-package-mo-id* is registered as subidentifier 176 under *management attribute*, and the attribute *r-af-ts-u-mo-id* is registered as subidentifier 115.

ANNEX D

MANAGED OBJECT CONTAINMENT

(This annex is part of the draft Recommendation.)

This annex depicts the allowed containment relationships of the various managed objects specified in this draft Recommendation. Refer to 4.3 for a general discussion of managed object containment.

The diagrams that are used to illustrate these containment relationships are organized according to two types of groupings. First, the managed objects that are common to all Service Agreements are presented. Second, managed object containment configurations are presented as the valid set of Service Package 'building blocks', such that any valid Service Package can be constructed by combining one or more of these building block configurations. Each building block configuration is illustrated both in terms of the

a) TransferService, ChannelStored, and Production managed objects that are to be contained by the ServicePackage managed object; and

b) corresponding SLE data channel managed objects that are contained by the DataChannelProfile managed object to support that Service Package configuration.

One containment configuration is depicted for each source of data for service packages that provide return SLE services, and for each sink of data for service packages that provide forward SLE services.

This annex is organized as follows:

a) Subsection D1 illustrates the containment relationships of the common managed objects defined in section 5.

b) Subsection D2 illustrates the containment relationships of the Return functional group managed objects defined in sections 7, 8, and 9.

c) Subsection D3 illustrates the containment relationships of the Forward Telecommand functional group managed objects defined in sections 10, 11, and 12.

This annex does not provide containment diagrams for the Forward AOS functional groups or their corresponding SLE data channels.

Subsections D2 through D9 contain two types of diagrams. The first type illustrates a containment relationship of the specific TransferService, ChannelStored, and Production derived classes corresponding to a particular service package configuration. The second type illustrates the particular SLE data channels that correspond to that service package configuration.

In each containment configuration diagram showing the objects contained by a ServicePackage managed object, the managed objects that belong to the same functional group are enclosed by a heavy-lined, round-cornered box that is labeled with the name of that functional group. The boxes that represent the ServicePackage and DataChannelProfile managed objects are illustrated by broken-line rectangles to indicate that these managed objects are part of the overall structure of the Service Agreement and not part of any functional group.

Each managed object rectangle is labeled with the name of the managed object and the number of instances that can be contained by the superior (containing) managed object. For example, (0..n) indicates that zero, one or more instances may be contained.

NOTES

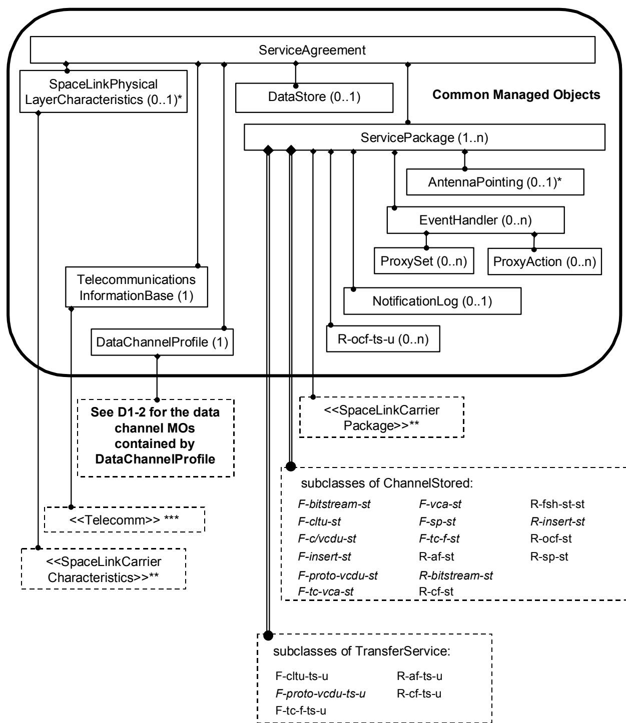
1) Italics in the figure below indicate that the managed object class is not specified, as CCSDS Panel 3 is not currently considering those services.

2) Double angle brackets (<< and >>) enclosing the name of a managed object class indicates a concrete derived class of the enclosed class. For instance, '<<SpaceLink-CarrierCharacteristics>>' means 'a concrete derived class of the SpaceLink-CarrierCharacteristics class'.

3) The diagrams in this annex show all allowed containment relationships. For typical cases, refer to reference [E3].

D1 Common Managed Objects

Figure D-1 illustrates the managed objects that are common to all service agreements. Since these managed objects form the root of any service agreement containment, all other managed objects in the service agreement are attached.



* These managed object classes only appear for service agreements that include space links.

** These managed object classes are defined in *Physical Layer,* reference [13].

*** These managed object classes are defined in *Mapping Rules,* reference [9].

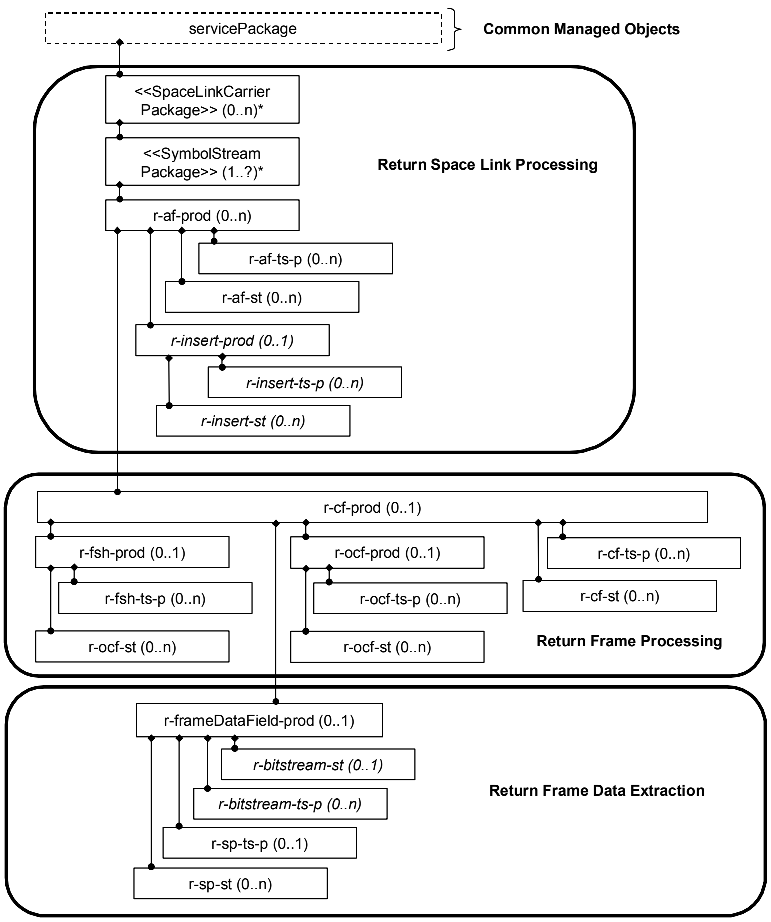
Figure D-1: Common Managed Object Classes

Double lines indicate that more than one managed object may be contained.

The managed objects contained by the ServiceAgreement and ServicePackage managed objects are too numerous to illustrate in figure D-1. The containment of the derived classes of the ChannelStored and TransferService managed objects are fully illustrated in the diagrams in subsections D2 and D3.

D2 Return Containment Configurations

Figure D-2 illustrates the managed objects corresponding to the resources needed to process services and storage functions where the return space link carrier is the source of the data to be handled. Note that the SpaceLinkCarrierPackage and SymbolStreamPackage managed objects, and the derived classes thereof, are specified in *Space Link Physical Layer Managed Object Specification*, reference [13]. The multiplicity of the SymbolStreamPackage containment by SpaceLinkCarrierPackage is labeled as unknown (‘?’) because this value depends upon the specific derived class.



* These managed object classes are defined in *Physical Layer,* reference [13].

Figure D-2: Containment by Space Link Carrier: servicePackage

Figure D-3 illustrates the SLE data channel managed objects that must be contained by the DataChannelProfile to support the configuration illustrated in figure D-2.

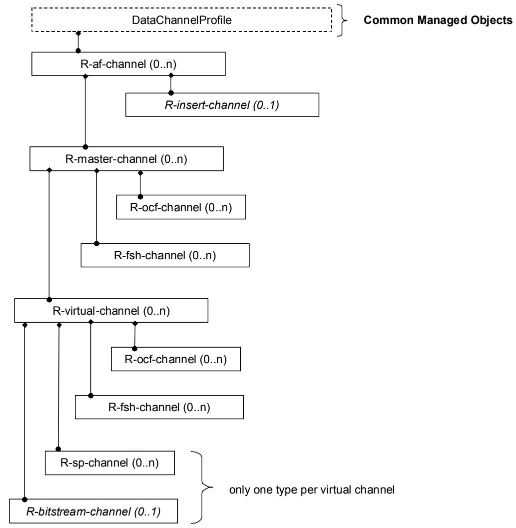


Figure D-3: Containment by Return Space Link Carrier, RAF Channel Storage, and RAF Service Used: DataChannelProfile

Figure D-4 illustrates the managed objects corresponding to the resources needed to provide services and storage functions where an RAF channel storage resource is the source of the data to be handled. The SLE data channel profile depicted in figure D-3 applies to this configuration as well.

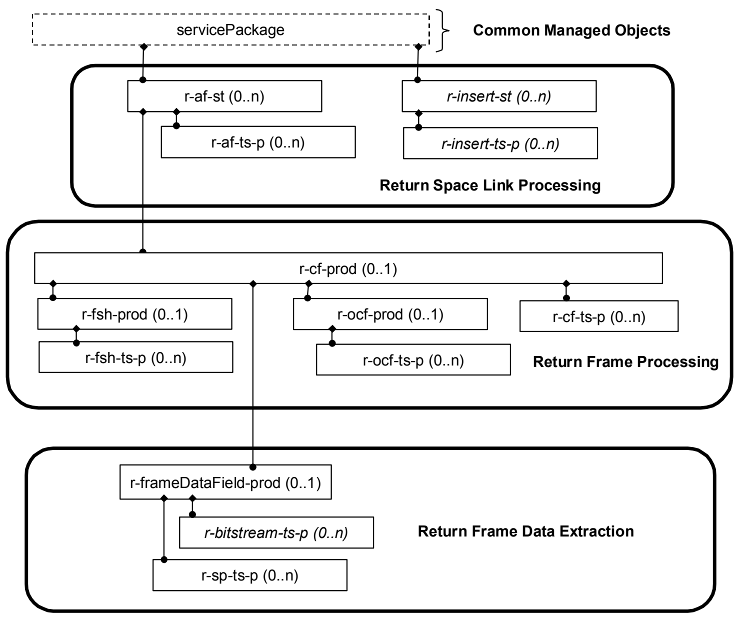


Figure D-4: Containment by RAF Channel Storage: servicePackage

Figure D-5 illustrates the managed objects corresponding to the resources needed to provide services and storage functions where an RAF service instance provided by another SLE Complex is the source of the data to be handled. The SLE data channel profile depicted in figure D-3 applies to this configuration as well.

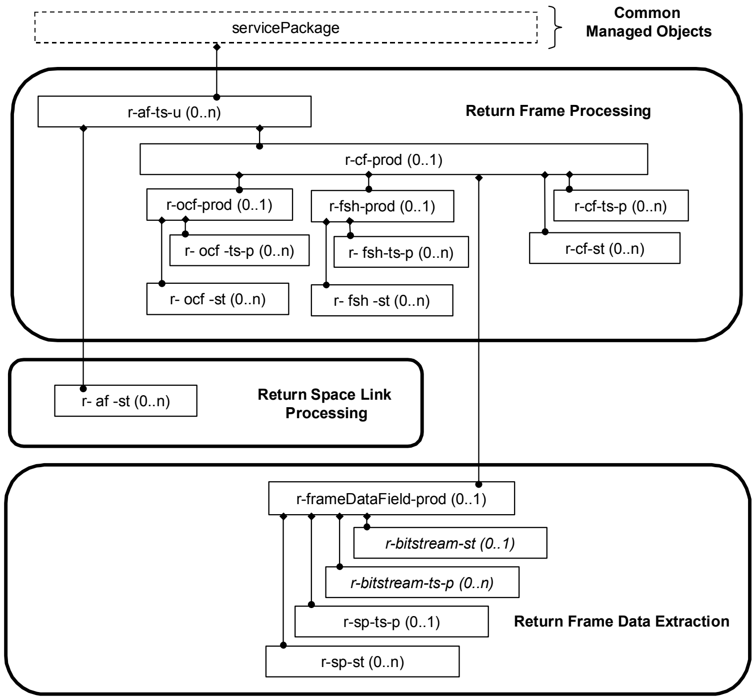


Figure D-5: Containment by R-af Service Used : servicePackage

Figure D-6 illustrates the managed objects corresponding to the resources needed to provide services and storage functions where an RCF channel storage resource is the source of the data to be handled. Figure D-7 illustrates the SLE data channel managed objects that must be contained by the DataChannelProfile to support the configuration illustrated in figure D-6.

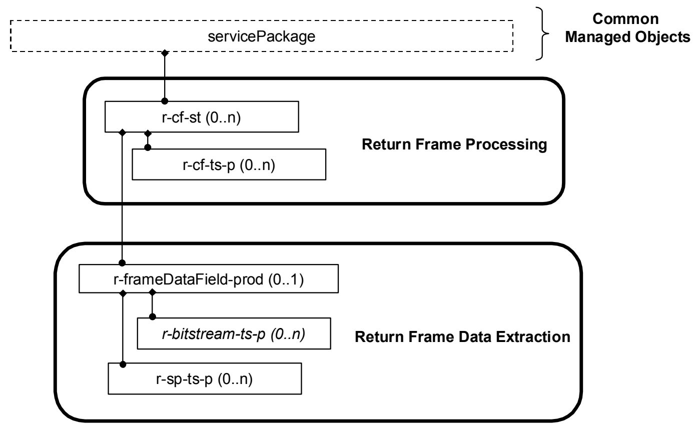


Figure D-6: Containment by RCF Channel Storage : servicePackage

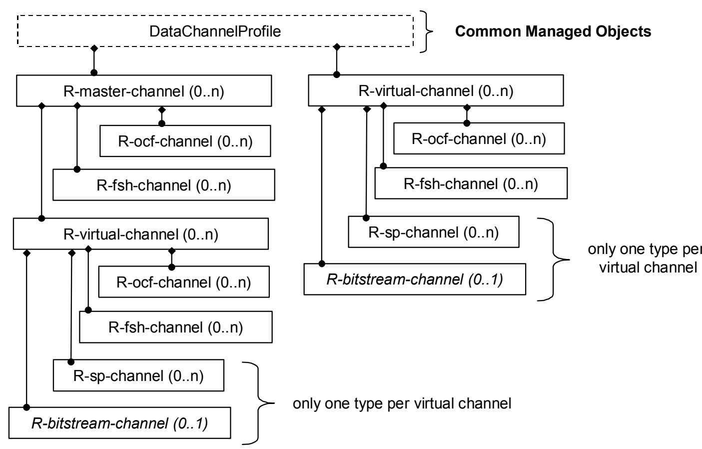


Figure D-7: Containment by RCF Channel Storage and RCF Service Used: DataChannelProfile

Figure D-8 illustrates the managed objects corresponding to the resources needed to provide services and storage functions where an RCF service instance provided by another SLE Complex is the source of the data to be handled. The SLE data channel profile depicted in figure D-7 applies to this configuration as well.

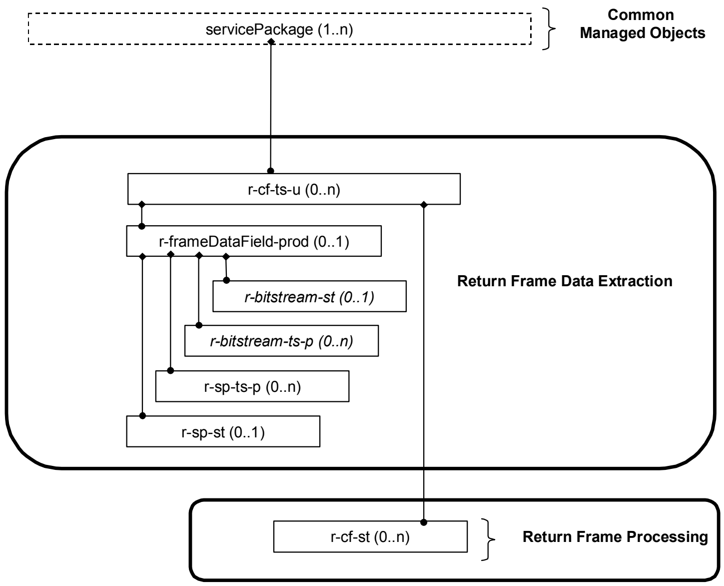


Figure D-8: Containment by RCF Service Used: servicePackage

Figure D-9 illustrates the managed objects corresponding to the resources needed to provide services and storage functions where R-Insert, R-OCF, R-FSH, R-SP, and/or R-Bitstream channel storage resources are the sources of the data to be handled. Figure D-10 illustrates the SLE data channel managed objects that must be contained by the DataChannelProfile to support the configuration illustrated in figure D-9.

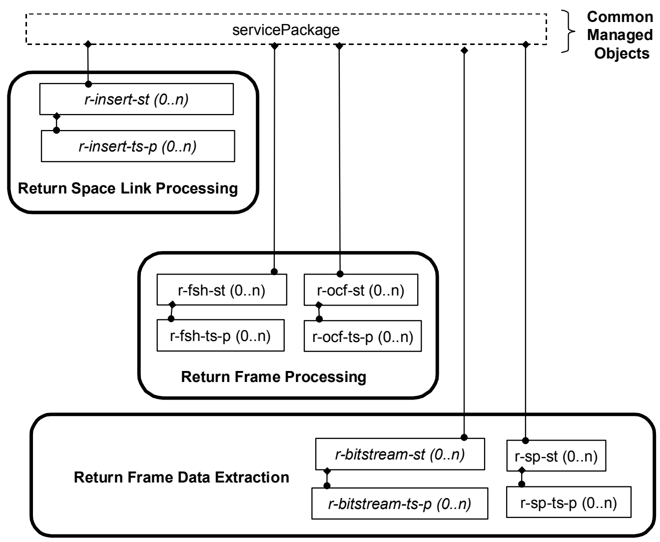


Figure D-9: Containment by R-Insert, R-OCF, R-FSH, R-SP, and R-Bitstream Channel Storage: servicePackage

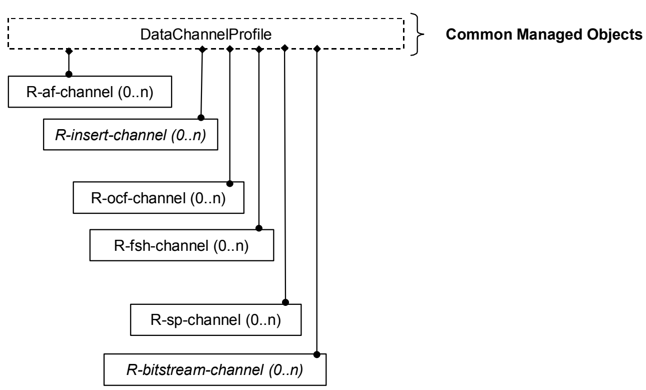


Figure D-10: Containment by R-Insert, R-OCF, R-FSH, R-SP, and R-Bitstream Channel Storage: DataChannelProfile

D3 Forward Telecommand Containment Configurations

Figure D-11 illustrates the managed objects corresponding to the resources needed to process services and storage functions where the forward space link carrier is the sink of the data being handled. Note that the SpaceLinkCarrierPackage and SymbolStreamPackage managed objects, and the derived classes thereof, are specified in *Space Link Physical Layer Managed Object Specification*, reference [13]. The multiplicity of the SymbolStreamPackage containment by SpaceLinkCarrierPackage is labeled as unknown ('?') because this value depends upon the specific derived class.

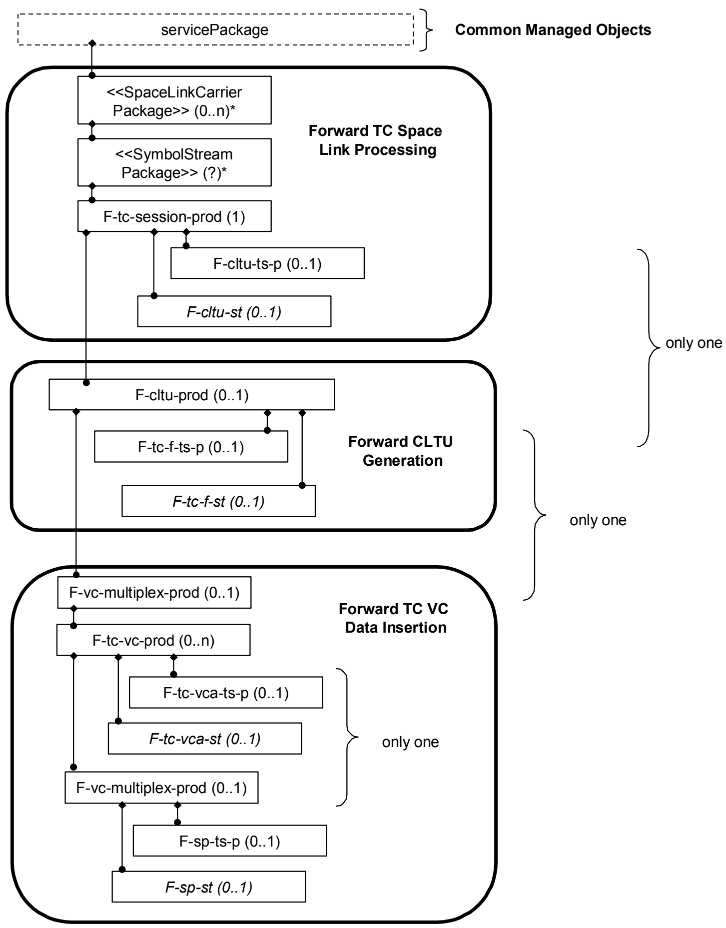


Figure D-11: Containment by Forward Space Link Carrier: servicePackage

Figure D-12 illustrates the SLE data channel managed objects that must be contained by the DataChannelProfile to support the configuration illustrated in figure D-11.

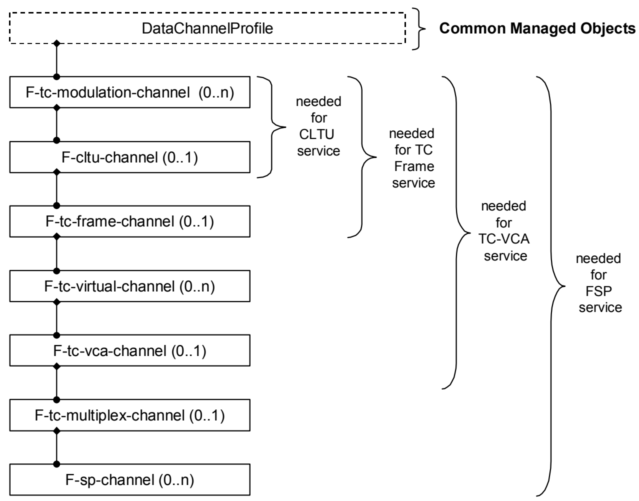


Figure D-12: Containment by Forward Space Link Carrier: DataChannelProfile

Figure D-13 illustrates the configuration of managed objects corresponding to the resources needed to provide services and storage functions where a CLTU channel storage resource is the sink of the data being handled. Figure D-14 illustrates the SLE data channel managed objects that must be contained by the DataChannelProfile to support the configuration illustrated in figure D-13. Note that because offline CLTU service is currently undefined, this configuration is currently unavailable.

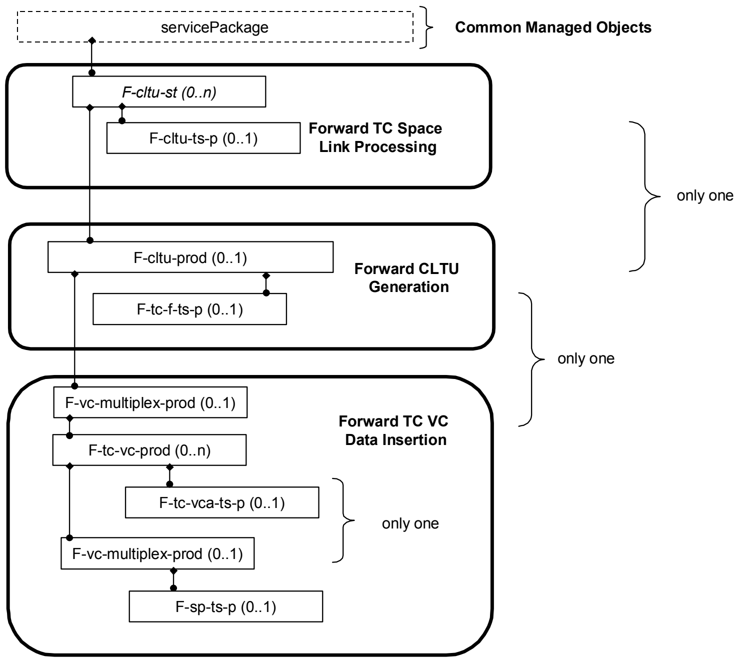


Figure D-13: Containment by Forward CLTU Channel Storage: servicePackage

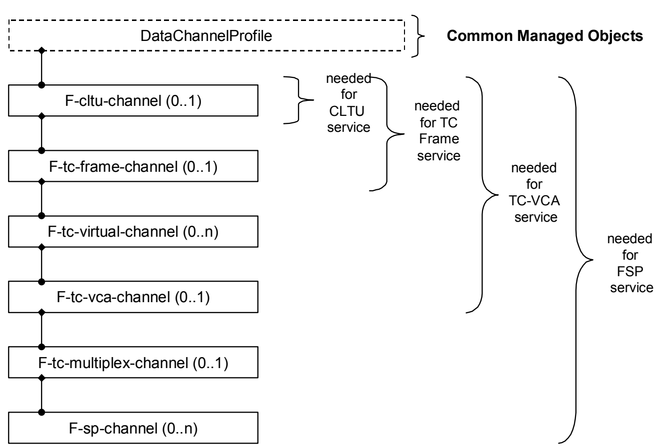


Figure D-14: Containment by Forward CLTU Channel Storage and CLTU Service Used: DataChannelProfile

Figure D-15 illustrates the managed objects corresponding to the resources needed to provide services and storage functions where a CLTU service instance provided by another SLE Complex is the sink of the data being handled. The SLE data channel profile depicted in figure D-14 applies to this configuration as well.

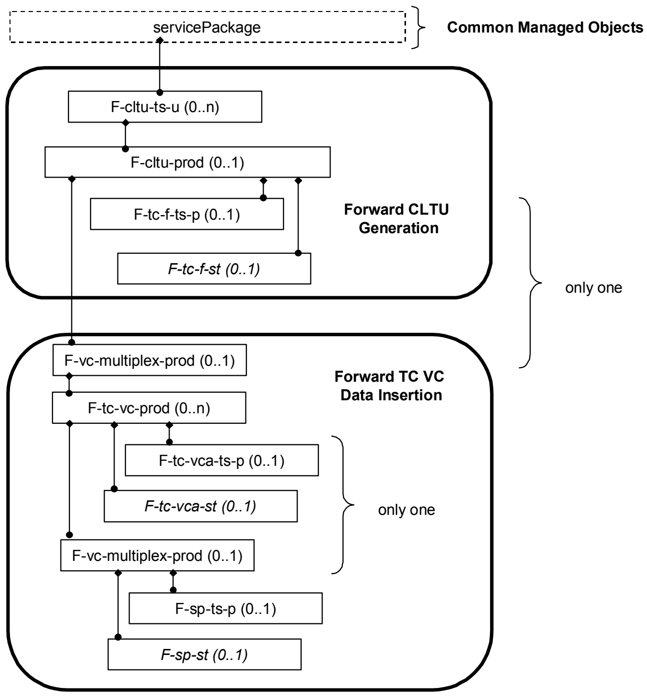


Figure D-15: Containment by Forward CLTU Service Used: servicePackage

Figure D-16 illustrates the managed objects corresponding to the resources needed to provide services and storage functions where a TC Frame channel storage resource is the sink of the data to be handled. Figure D-17 illustrates the SLE data channel managed objects that must be contained by the DataChannelProfile to support the configuration illustrated in figure D-16. Note that because offline TC Frame service is currently undefined, this configuration is currently unavailable.

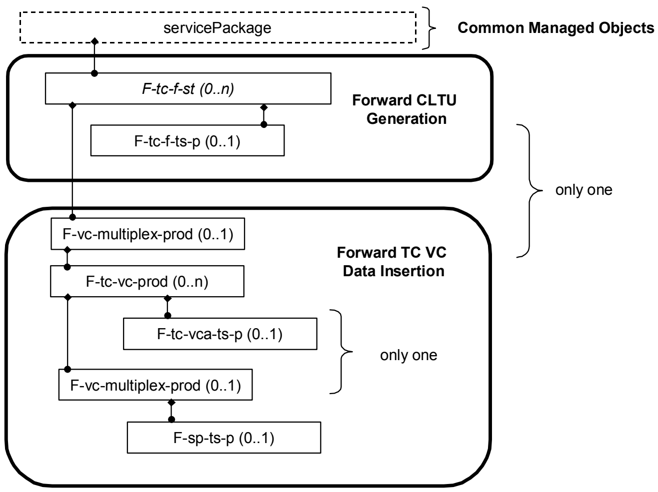


Figure D-16: Containment by Forward TC Frame Channel Storage: servicePackage

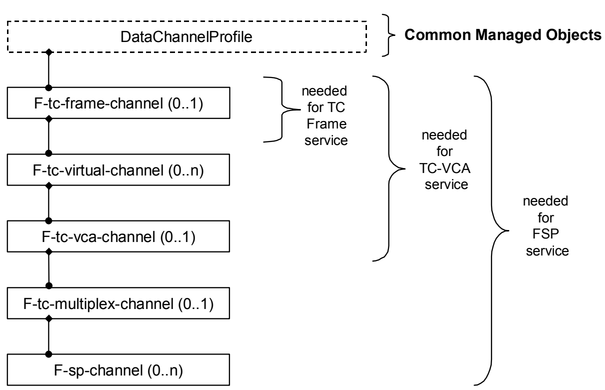


Figure D-17: Containment by Forward TC Frame Channel Storage and TC Frame Service Used: DataChannelProfile

Figure D-18 illustrates the managed objects corresponding to the resources needed to provide services and storage functions where a TC Frame service instance provided by another SLE Complex is the source of the data to be handled. The SLE data channel profile depicted in figure D-17 applies to this configuration as well.

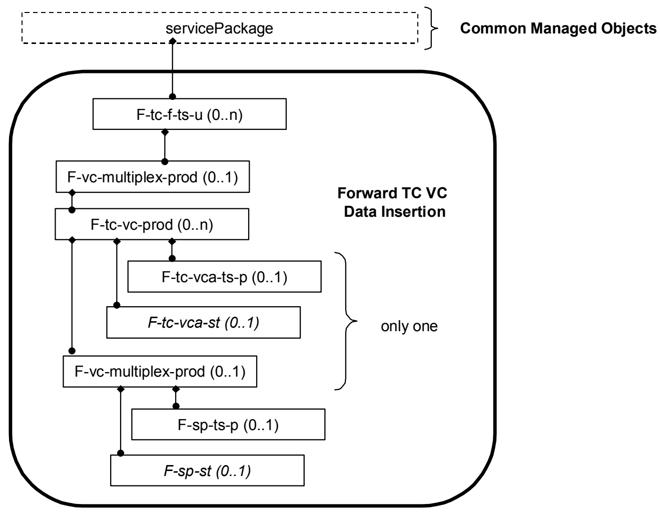


Figure D-18: Containment by Forward TC Frame Service Used: servicePackage

Figure D-19 illustrates the managed objects corresponding to the resources needed to provide services and storage functions where TC-VCA or FSP channel storage resources are the sinks of the data to be handled. Figure D-20 illustrates the SLE data channel managed objects that must be contained by the DataChannelProfile to support the configuration illustrated in figure D-19. Note that because offline TC-VCA and FSP services are currently undefined, this configuration is currently unavailable.

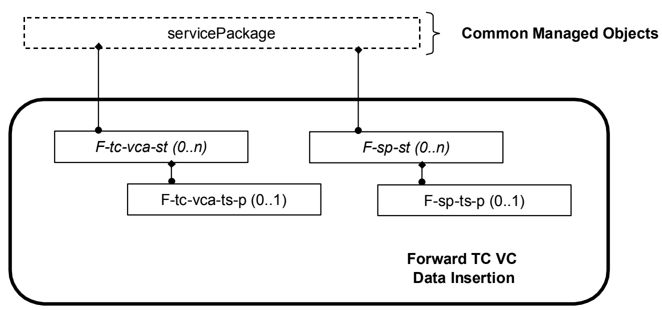


Figure D-19: Containment by Forward TC-VCA and/or FSP Channel Storage: servicePackage

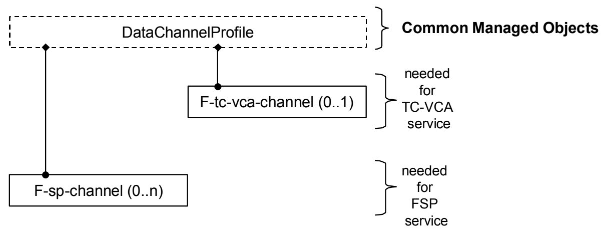


Figure D-20: Containment by Forward TC-VCA and FSP Channel Storage: DataChannelProfile

ANNEX E

INFORMATIVE REFERENCES

[E1] *Procedures Manual for the Consultative Committee for Space Data Systems*. CCSDS A00.0-Y-7. Yellow Book. Issue 7. Washington, D.C.: CCSDS, November 1996.

[E2] *Radio Frequency and Modulation Systems—Part 1: Earth Stations and Spacecraft*. Recommendation for Space Data System Standards, CCSDS 401.0-B. Blue Book. Washington, D.C.: CCSDS, May 1999.

[E3] *Cross Support Concept—Part 1: Space Link Extension Services*. Report Concerning Space Data Systems Standards, CCSDS 910.3-G-1. Green Book. Issue 1. Washington, D.C.: CCSDS, May 1995.

[E4] *Guidelines for the Definition of Managed Objects.* ISO/IEC 10165-4 (x.722). Geneva: ISO, 1992.

[E5] *Information Technology—Open Systems Interconnection—Basic Reference Model Part 4: Management Framework.* ISO/IEC 7498-4. Geneva: ISO, 1989.

[E6] *InformationTechnology—Open Systems Interconnection—The Directory: Models* ISO/IEC 9594-2:1995 (E). Geneva: ISO, 1995.

[E7] *Information Technology—Open Systems Interconnection—The Directory—Part 8: Authentication Framework.* ISO/IEC 9594-8. Geneva: ISO, 1993.

[E8] *Unified Modeling Language Reference Manual (Addison-Wesley Object Technology Series)*. James Rumbaugh, Grady Booch, and Ivar Jacobson. New York: Addison-Wesley, 1998.

[E9] *Object-Oriented Analysis.* Peter Coad and Edward Yourdon (Contributor). New York: Yourdon Press Computing Series, 1991.

[E10] *SLE Service Specifications*. A group of documents in various stages of production that specify the SLE transfer services. The services include (but are not limited to):

Return All Frames	Forward CLTU
Return Channel Frames	Forward Space Packets
Return Frame Secondary Header	Forward TC Frames
Return Operational Control Field	Forward TC VCA
Return Space Packets	

[E11] *Standard Formatted Data Units—Control Authority Procedures*. Recommendation for Space Data System Standards, CCSDS 630.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, June 1993.

[E12] *SNMP, SNMPv2, and CMIP: The Practical Guide to Network Management Standards*. William Stallings. New York: Addison-Wesley, 1993.